INFORMATION AND COMPUTATION **87**, 2–57 (1990)

A Compositional Proof System on a Category of Labelled Transition Systems

GLYNN WINSKEL

Computer Science Department, Aarhus University,
Ny Munkegade, 8000 Aarhus C, Denmark

This paper presents an attempt to cast labelled transition systems, and other models of parallel computation, in a category-theoretic framework. One aim is to use category theory to provide abstract characterisations of constructions like parallel composition valid throughout a range of different models and to provide formal means for translating between different models. Another aim is to exploit the framework of categorical logic to systematise specification languages and the derivation of proof systems for parallel processes. After a category of labelled transition systems is presented, its categorical constructions are used to establish a compositional proof system. A category of properties of transition systems indexed by the category of labelled transition systems is used in forming the proof system.

INTRODUCTION

I think there has been a fair degree of success in understanding and relating different models of concurrency by placing them in a category-theoretic framework. By viewing each class of models (transition systems form one such class of models) as a category, obtained by providing it with a suitable notion of morphism, the relationship between models can often be expressed as an adjunction between categories. Because of the way the right adjoints preserve limits and left adjoints colimits, this leads to a smooth translation between semantics in terms of one model and semantics in terms of another. The morphisms make sense intuitively and represent a "partial simulation" of one process in another. (See, e.g., Winskel, 1984, for a survey.)

It is clear that the nature of the events determines the nature of parallel compositions; for instance, in Milner's (1980, 1989) CCS we only want synchronisations between events which are complementary in the sense that one is an α and the other an $\bar{\alpha}$ event. The labelling of events to specify their nature has never been incorporated convincingly into the category-theoretic setup just described, though there have been at least two attempts

0890-5401/90 $3.00

(Winskel, 1982; Labella, 1985). Without extending morphisms in the models to account for labels operations like restriction, relabelling, and various parallel compositions are not truly categorical, which obstructs the category-theoretic account of parallel computation. This restricts the use that can be made of techniques from category theory. I have in mind categorical logic especially, which, at the very least, provides guidelines for the construction of models and proof systems. A recent success, exploiting ideas from categorical logic, is the work of Abramsky (1987) on a logic of domains.

This paper presents a way to understand constructions on parallel processes in the terms of indexed category theory. In particular it uses an indexed category of processes to explain the important operations of restriction and relabelling categorically and, through these, give a category-theoretic account of parallel compositions. Building on this work, it presents a further application of indexed category theory; a category of properties indexed by processes is used to describe a logic to reason about parallel processes in a structured way.

At present, we do not have the general notion of a categorical model for parallel processes. Although the results here often hold for a range of different models we must work with a typical model as an example. Besides, given our incomplete understanding of the different ways of handling parallelism, this concreteness is all for the better. For the main model I have chosen labelled transition systems, though synchronisation trees play a role, and the treatment of other models is referred to briefly.[1] A definition of morphism on labelled transition systems is given. Its categorical properties are described in the language of indexed category theory, the idea being that a transition system is indexed by its set of labels. This indexed category yields familiar constructions suitable for interpreting a language of parallel processes. Analogous results hold for an indexed category of synchronisation trees. The indexed categories, labelled transition systems and synchronisation trees, are related by an adjunction which respects the indexing. This provides a smooth translation between the two models and instant proofs that semantics is respected in passing from one model to the other. Supplied with a definition of morphism between transition systems and a language of terms for labelled transition systems based on categorical combinators, we investigate how modal assertions are preserved and reflected by morphisms and finally present a compositional proof system whose form is guided by a category of properties of transition systems.

The work uses a particular representation of partial functions, the details and notation for which are found in Appendix I.

[1] See Winskel (1988) for a similar presentation with Petri nets as the model.

1. A CATEGORY OF LABELLED TRANSITION SYSTEMS

Labelled transition systems are a frequently used model of parallel processes. They consist of a set of states, with an initial state, together with transitions between states which are labelled to specify the kind of events they represent.

1.1. DEFINITION. A *labelled transition system* is a structure

$$(S, i, L, Tran),$$

where

S is a set of states with *initial state* i,

L is a set of labels, always assumed to not contain a distinguished symbol $*$,

$Tran \subseteq S \times L \times S$ is the transition relation.

This definition narrows attention to labelled transition systems, which are *extensional* in that they cannot have two distinct transitions with the same label and the same pre- and poststates.

1.2. *Notation.* Let $(S, i, L, Tran)$ be a labelled transition system. We write

$$s \xrightarrow{\alpha} s'$$

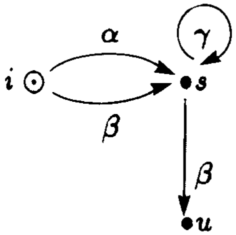
to indicate that $(s, \alpha, s') \in Tran$, and

$$s \longrightarrow s'$$

when $\exists \alpha.\, s \xrightarrow{\alpha} s'$. A state s is said to be *reachable* when

$$i \longrightarrow^* s.$$

This notation lends itself to the familiar graphical notation for labelled transition systems. For example,



represents a labelled transition system which at the initial state i can perform either an α or a β transition to enter the state s at which it can repeatedly perform a γ transition or a β transition to enter state u.

When t is a transition of the form $t=(s, \alpha, s')$ we sometimes use the notation ${}^{\bullet}t$ for the prestate s and $t^{\bullet}$ for the poststate s'.

It is technically convenient to introduce *idle* transitions, associated with any state.

1.3. Definition. Let $T=(S, i, L, Tran)$ be a labelled transition system. An *idle transition* of T consists of $(s, *, s)$ for $s \in S$. Define

$$Tran_{*} = Tran \cup \{(s, *, s) \mid s \in S\}.$$

Idle transitions play a role in the definition of a morphism between labelled transition systems.

1.4. Definition. Let

$$T_0 = (S_0, i_0, L_0, Tran_0)$$

and

$$T_1 = (S_1, i_1, L_1, Tran_1)$$

be labelled transition systems. A *morphism* $f\colon T_0 \to T_1$ is a pair $f=(\sigma, \lambda)$, where

$$\sigma : S_0 \to S_1$$

$$\lambda : L_0 \to_{*} L_1$$

are such that $\sigma(i_0)=i_1$ and

$$(s, \alpha, s') \in Tran_0 \Rightarrow (\sigma(s), \lambda(\alpha), \sigma(s')) \in Tran_{1*}.$$

In particular, say a morphism (σ, λ) of transition systems is *label preserving* when λ is the identity function on a set of labels.

With the introduction of idle transitions, morphisms on labelled transition systems can be described as preserving transitions and the initial state. Observe that morphisms between labelled transition systems can be characterised in a way which does not involve idle transitions. According to this characterisation a morphism between two labelled transition systems $T_0=(S_0, i_0, L_0, Tran_0)$ and $T_1=(S_1, i_1, L_1, Tran_1)$ consists of (σ, λ), where $\sigma: S_0 \to S_1$ and $\lambda: L_0 \to_{*} L_1$, which satisfy

$\sigma(i_0)=i_1$

$(s, \alpha, s') \in Tran_0$ & $\lambda(\alpha)$ defined $\Rightarrow (\sigma(s), \lambda(\alpha), \sigma(s')) \in Tran_1$, and

$(s, \alpha, s') \in Tran_0$ & $\lambda(\alpha)$ undefined $\Rightarrow \sigma(s)=\sigma(s')$.

As this characterisation makes clear, the intention behind the definition of morphism is that the effect of a transition with label α in T_0 leads to inaction in T_1 precisely when $\lambda(\alpha)$ is undefined. In our definition of morphism, idle transitions represent this inaction, a device which avoids the fuss of considering whether or not $\lambda(\alpha)$ is defined. It is to be stressed that an idle transition $(s, *, s)$ represents inaction and is to be distinguished from the action expressed by a transition (s, α, s') for a label α.

Morphisms preserve initial states and transitions and so clearly preserve reachable states:

1.5. PROPOSITION. *Let $(\sigma, \lambda)\colon T_0 \to T_1$ be a morphism of labelled transition systems. Then if s is a reachable state of T_0 then $\sigma(s)$ is a reachable state of T_1.*

Labelled transition systems with morphisms as defined form a category which will be central to our study:

1.6. PROPOSITION. *Labelled transition systems with morphisms form a category in which the composition of two morphisms $f = (\sigma, \lambda)\colon T_0 \to T_1$ and $g = (\sigma', \lambda')\colon T_1 \to T_2$ is $g \circ f = (\sigma' \circ \lambda)\colon T_0 \to T_2$ and the identity morphism for a transition system T has the form $(1_S, 1_L)$, where 1_S is the identity function on states and 1_L is the identity function on the labelling set L of T.*

(Here composition on the left of a pair is that of total functions while composition on the right is that of partial functions.)

1.7. DEFINITION. Denote by **T** the category of labelled transition systems given by the last proposition.

2. LABELLED TRANSITION SYSTEMS AS AN INDEXED CATEGORY

Restriction and relabelling are important operations on processes. For example, in Milner's CCS, labels are used to distinguish between input and output to channels, connected to processes at ports, and internal events. The effect of hiding all but a specified set of ports of a process, so that communication can no longer take place at the hidden ports, is to restrict the original behaviour of the process to transitions which do not occur at the hidden ports. Given a labelled transition system and a subset of its labelling set, the operation of restriction removes all transitions whose labels are not in that set. In CCS, one can make copies of a process by renaming its port names. This is associated with the operation of relabelling the transitions in the labelled transition system representing its behaviour.

Restriction and relabelling are constructions which depend on labelling sets and functions between them. Seeing them as categorical constructions involves dealing explicitly with functions on labelling sets and borrowing a couple of fundamental ideas from indexed category theory. An indexed category is, naturally enough, a collection of categories indexed functorially by a base category. There is an alternative presentation of the same idea proposed originally by Grothendieck and argued for forcibly by Bénabou For them the central concept is that of a fibration, of which the category **T** of labelled transition systems is an example.

2.1. Definition. Let $p: \mathbf{X} \to \mathbf{B}$ be a functor.

A morphism $f: X \to X'$ in **X** is said to be *cartesian* with respect to p if for any $g: X'' \to X'$ in **X** and morphism $\sigma: p(X'') \to p(X)$ in **B** for which $p(f) \circ \sigma = p(g)$ there is a unique morphism $h: X'' \to X$ such that $p(h) = \sigma$ and $f \circ h = g$. A cartesian morphism $f: X \to X'$ in **X** is said to be a *cartesian lifting* of the morphism $p(f)$ in **B** with respect to X'. Say $p: \mathbf{X} \to \mathbf{B}$ is a *fibration* if every morphism $\sigma: B \to B'$ in **B** has a cartesian lifting with respect to any X' such that $p(X') = B'$. A morphism $f: X \to X'$ in **X** is said to be *vertical* if $p(f) = 1_{p(X)}$.

Often p is called the *projection*, **B** the *base category*, and each full subcategory $p^{-1}(B)$ of **X**, which is sent to the subcategory consisting of the identity morphism on an object B of **B**, the *fibre* over B.

There is a projection functor $p: \mathbf{T} \to \mathbf{Set}_*$, to sets with partial functions, which sends a morphism of labelled transition systems $(\sigma, \lambda): T \to T'$ between labelled transition systems T over L and T' over L' to the partial function $\lambda: L \to_* L'$. The projection $p: \mathbf{T} \to \mathbf{Set}_*$ determines a fibration. Let us represent what this means in a diagram. Suppose $\lambda: L \to_* L'$ is a partial function and that T' is a labelled transition system with $p(T') = L'$, i.e., with labelling set L'. We require a labelled transition system T and a cartesian morphism $f: T \to T'$ such that $p(f) = \lambda$. Assuming $f': T'' \to T$ is a morphism such that $p(f') = \lambda': L'' \to_* L$ and $\lambda'': L'' \to_* L'$ with $\lambda \circ \lambda'' = \lambda'$, this means there is a unique morphism $f'': T'' \to T$ such that $p(f'') = \lambda''$ and $f \circ f'' = f'$:

$$\begin{tikzcd}
T'' \arrow[d, "f''"'] \arrow[dr, "f'"] & \\
T \arrow[r, "f"] & T'
\end{tikzcd}$$

$$\begin{tikzcd}
L'' \arrow[d, "\lambda''"', "*" near end] \arrow[dr, "\lambda'", "*"' near end] & \\
L \arrow[r, "\lambda", "*"' near end] & L'
\end{tikzcd}$$

The following definition gives the explicit construction of the cartesian lifting, with the labelled transition system T given by the operation $\lambda^*(T')$.

2.2. Definition. Let $\lambda: L \rightarrow_* L'$. Let $T' = (S, i, L', \mathit{Tran}')$ be a labelled transition system. Define $\lambda^*(T') = (S, i, L, \mathit{Tran})$, where

$$\mathit{Tran} = \{(s, \alpha, t) \mid (s, \lambda(\alpha), t) \in \mathit{Tran}'_*\}.$$

2.3. Proposition. *Let $\lambda: L \rightarrow_* L'$ be a partial function on label sets. Let $T' = (S, i, L', \mathit{Tran}')$ be a labelled transition system. Using the notation of the definition above, $\lambda^*(T') = (S, i, L, \mathit{Tran})$ is a labelled transition system. The pair $(1_S, \lambda)$ is a cartesian morphism $\lambda^*(T') \rightarrow T'$ with respect to the projection $p: \mathbf{T} \rightarrow \mathbf{Set}_*$.*

Proof. Let $\lambda: L \rightarrow_* L'$ be a partial function between sets of labels. Let $T' = (S, i, L', \mathit{Tran}')$ be a labelled transition system. Let $\lambda^*(T') = (S, i, L, \mathit{Tran})$ and $f = (1_S, \lambda)$ as defined above. Certainly $\lambda^*(T')$ is a labelled transition system and $f: \lambda^*(T') \rightarrow T'$ is a morphism of labelled transition systems. We require that f be cartesian.

To this end, suppose

$$\lambda': L'' \rightarrow_* L' \qquad \text{and} \qquad \lambda'': L'' \rightarrow L$$

with $\lambda \circ \lambda'' = \lambda'$ and that $f': T' \rightarrow T''$ is a morphism of labelled transition systems such that $p(f') = \lambda'$. We require the existence of a unique morphism $f'': T'' \rightarrow \lambda^*(T')$ such that $p(f'') = \lambda''$ and $f \circ f'' = f'$. However, f' must have the form $f' = (\sigma', \lambda')$ for a function $\sigma': S' \rightarrow S''$. If i'' is the initial state of T'' then $\sigma'(i'') = i'$, the initial state of $\lambda^*(T')$. Consider (s, α, t), a transition of T''. Then as f' in a morphism we see $(\sigma'(s), \lambda'(\alpha), \sigma'(t)) \in \mathit{Tran}'_*$, so $(\sigma'(s), \lambda \circ \lambda''(\alpha), \sigma'(t)) \in \mathit{Tran}'_*$. But this implies $(\sigma'(s), \lambda''(\alpha), \sigma'(t)) \in \mathit{Tran}_*$. This ensures that (σ', λ'') is the unique morphism $f'': T'' \rightarrow \lambda^*(T')$ such that $p(f'') = \lambda''$ and $f \circ f'' = f'$. Hence f is cartesian. ▮

As a corollary we obtain:

2.4. Theorem. *The projection $p: \mathbf{T} \rightarrow \mathbf{Set}_*$ determines a fibration.*

The construction of a cartesian lifting becomes simpler and familiar for special cases of λ. If λ is an inclusion $\lambda: L \subseteq L'$ and T' has labelling set L', then $\lambda^*(T')$ is just the restriction of T' to transitions with labels in L.

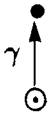
2.5. Definition. Let $T = (S, i, L', \mathit{Tran}')$ be a labelled transition system. Let $L \subseteq L'$. Define the *restriction* $T \lceil L$ to be the transition system (S, i, L, Tran) with

$$\mathit{Tran} = \{(s, \alpha, t) \in \mathit{Tran}' \mid \alpha \in L\}.$$

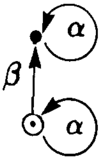
2.6. PROPOSITION. *Let $T=(S, i, L', Tran')$ be a labelled transition system. Let $L \subseteq L'$ and $j: L \hookrightarrow L'$ be the subset morphism. Then $T \lceil L$ is a transition system and $(1_S, j): T \lceil L \rightarrow T$ is cartesian with respect to the projection p from the labelled transition system to their sets of labels.*

Let $\lambda: L \rightarrow_* L'$ and T' be a labelled transition system with labelling set L'. If λ is 1–1 and total then $\lambda^*(T')$ is simply a relabelling of the transitions of T'. Such constructions are well known from work on CCS and CSP. More novel are the constructions when λ is really partial or sends two distinct labels to the same label. If λ is undefined on a label α, then the transitions of $\lambda^*(T')$ include all transitions of the form (s, α, s), where s is a state. These loops at any state introduce an "α stutter" into the transitions of $\lambda^*(T')$. If λ takes several distinct labels $\{\alpha_j \mid j \in J\}$ to a common label γ, any transition (s, γ, s'), with label γ, in T' is replaced in $\lambda^*(T')$ by copies, (s, α_j, s'), one for each $j \in J$. These cases are illustrated by the following examples. While not directly associated with familiar programming constructs like restriction and relabelling, cartesian morphisms like those in the example will play a role in deriving parallel compositions from the fibre product (see 3.3).

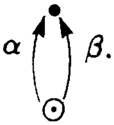
2.7. EXAMPLE. Let $\lambda_0: \{\alpha, \beta\} \rightarrow_* \{\gamma\}$ be the partial function so $\lambda_0(\beta)=\gamma$ and $\lambda_0(\alpha)$ is undefined. Let T be the transition system



Then $\lambda_0^*(T)$ is the transition system



which can "stutter" on α, in that every state can repeatedly perform an α transition. Let $\lambda_1: \{\alpha, \beta\} \rightarrow_* \{\gamma\}$ be total so that $\lambda_1(\alpha)=\lambda_1(\beta)=\gamma$. This time $\lambda_1^*(T)$ is the transition system



We return to consider the fibration $p: \mathbf{T} \to \mathbf{Set}_*$. Each fibre $p^{-1}(L)$ is a subcategory of labelled transition systems over a set of labels L whose labelling functions project to the identity function on L. In our proof that p does indeed determine a fibration, a particular choice of cartesian liftings was given; for $\lambda: L \to_* L'$ and $T' \in p^{-1}(L')$ the morphism $c(\lambda, T') = (1_S, \lambda): \lambda^*(T') \to T'$, based on the particular construction $\lambda^*(T')$ and involving the states S' of T', was shown to be cartesian. In general, such a function c making a choice of cartesian lifting for a fibration is called a *cleavage*. The fibration ensures that two cleavages are the same to within vertical isomorphism; if $c(\lambda, T')$ and $c'(\lambda, T')$ are two choices for the cartesian lifting of λ with respect to T' then there is a unique vertical isomorphism θ such that $c'(\lambda, T') = c(\lambda, T') \circ \theta$. A cleavage for a fibration specifies a functor between fibres, a general argument which we carry out for the fibration p of labelled transition systems. Define the functor $\lambda^*: p^{-1}(L') \to p^{-1}(L)$ for each partial function $\lambda: L \to_* L'$ as follows:

Let $f': T_0' \to T_1'$ be a morphism in the fibre $p^{-1}(L')$. The cleavage specifies cartesian morphisms

$$c(\lambda, T_0'): \lambda^*(T_0') \to T_0' \qquad \text{and} \qquad c(\lambda, T_1'): \lambda^*(T_1') \to T_1'.$$

As the morphism $c(\lambda, T_1')$ is cartesian, the composition $f'c(\lambda, T_0')$ factors as $c(\lambda, T_1')\, f$ for some unique $f: \lambda^*(T_0) \to \lambda^*(T_1)$, such that $p(f) = 1_L$. We extend λ^* to act on morphisms like f' by taking $\lambda^*(f') = f$. It can be checked that we obtain a functor in this way.

The precise functor is specified by the cleavage for p. However, even without these extra data, the functor is determined to within natural isomorphism by the fact that p is a fibration. It is simple to give a direct characterisation of the functor λ^*.

2.8. PROPOSITION. *Let $\lambda: L \to_* L'$ be a partial function between sets of labels L, L'. Let $f' = (\sigma', 1_{L'}): T_0' \to T_1'$ be a morphism in the fibre $p^{-1}(L')$. Then*

$$\lambda^*(f') = (\sigma', 1_L).$$

We have seen how restriction arises as a special case of the operation λ^* when λ is an inclusion. A general relabelling construction is obtained from an instance of a dual notion, that of a cofibration. At least when $\lambda: L \to_* L'$ is total, there is an obvious functor $\lambda_!: p^{-1}(L) \to p^{-1}(L')$ which takes a labelled transition system T to $\lambda_!(T)$, the same underlying transition system but relabelled according to λ. There is a morphism $(1_S, \lambda): T \to \lambda_!(T)$. In fact such a morphism satisfies a concept dual to that of being a cartesian lifting, and taking advantage of the fact that such relabelling functions can also be defined when λ is partial, we obtain a cofibration as now defined. This time the cofibration provides us with the functor $\lambda_!: p^{-1}(L) \to p^{-1}(L')$ between fibres.

2.9. Definition. Let $p: \mathbf{X} \to \mathbf{B}$ be a functor. It is a *cofibration* if $p^{\mathrm{op}}: X^{\mathrm{op}} \to \mathbf{B}^{\mathrm{op}}$ is a fibration. A morphism $f: X \to X'$ in $\mathbf{X}$ is said to be *cocartesian* with respect to p if f^{op} is cartesian in the fibration; the morphism f is a *cocartesian lifting* of $p(f)$.

Again it is helpful to explain with the help of a diagram. Suppose $\lambda: L \to_* L'$ is a partial function and that T is a labelled transition system with $p(T) = L$. We require a labelled transition system T' and a cocartesian morphism $f: T \to T'$ such that $p(f) = \lambda$. Assuming $f': T \to T''$ is a morphism such that $p(f') = \lambda': L \to_* L''$ and $\lambda'': L' \to_* L''$ with $\lambda'' \circ \lambda = \lambda'$, this means there is a unique morhism $f'': T' \to T''$ such that $p(f'') = \lambda''$ and $f'' \circ f = f'$:

$$\begin{tikzcd}
 & T'' \\
T \arrow[ur, "f'"] \arrow[r, "f"] & T' \arrow[u, "f''"']
\end{tikzcd}$$

$$\begin{tikzcd}
 & L'' \\
L \arrow[ur, "\lambda'", "*"' near end] \arrow[r, "\lambda", "*"' near end] & L' \arrow[u, "\lambda''"', "*" near end]
\end{tikzcd}$$

Here is the construction of the cocartesian lifting for labelled transition systems:

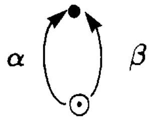
2.10. Definition. Let $\lambda: L \to_* L'$ be a partial function on label sets. Let $T = (S, i, L, Tran)$ be a labelled transition system. For $s, t \in S$ define

$$s \sim_1 t \text{ iff } \exists \beta \in L.(s, \beta, t) \in Tran \quad \& \quad \lambda(\beta) \text{ undefined.}$$

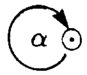
Let $\sim$ be the equivalence relation generated by $\sim_1$. Define $\lambda_!(T) = (S', i', L', Tran')$, where

$S' = \{\{s\}_\sim \mid s \in S\}$, the equivalence classes of S, with $i' = \{i\}_\sim$, and $Tran' = \{(\{s\}_\sim, \lambda(\alpha), \{t\}_\sim) \mid (s, \alpha, t) \in Tran \;\&\; \lambda(\alpha) \text{ defined}\}$.

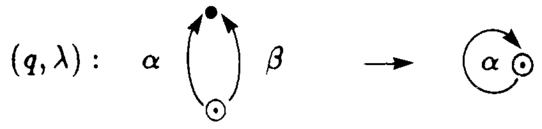
2.11. Example. Let $\lambda: \{\alpha, \beta\} \to_* \{\alpha\}$ be the partial function such that $\lambda(\alpha) = \alpha$ and $\lambda(\beta)$ is undefined. Let T be the labelled transition system



The two states are equivalent under the relation above with respect to our choice of λ, so that $\lambda_!(T)$ is the labelled transition system



Letting q be the associated quotient map on states, there is a morphism



which is, in fact, cocartesian with respect to the projection $p: \mathbf{T} \to \mathbf{Set}_*$ (see Proposition 2.12, below). Note how, because of the partiality of λ, we are forced to identify states when forming $\lambda_!(T)$ so that there is a morphism from T to $\lambda_!(T)$ which projects to λ under p.

For the general construction, when $\lambda: L \to L'$ may be partial, we are forced to identify states in order to obtain a morphism which is a cocartesian lifting. However, when λ is total, for a labelled transition system $T = (S, i, L, Tran)$, the construction simplifies to give $\lambda_!(T) = (S, i, L', Tran')$, where

$$(s, \beta, t) \in E' \Leftrightarrow \exists \alpha \cdot \beta = \lambda(\alpha) \qquad \& \qquad (s, \alpha, t) \in Tran.$$

(I have taken the liberty of identifying a state s with its equivalence class under the identity relation.)

This relabelling construction is the kind one expects and sees in transition-system semantics of languages like CCS and CSP.

2.12. Proposition. *Let $\lambda: L \to_* L'$ be a partial function on label sets. Let $T = (S, i, L, Tran)$ be a labelled transition system. Using the notation of the definition above, $\lambda_!(T) = (S', i', L', Tran')$ is a labelled transition system. Taking $q: S \to S'$ to be the map taking any state s to its equivalence class $\{s\}_\sim$, $(q, \lambda): T \to \lambda_!(T)$ is cocartesian with respect to the projection $p: \mathbf{T} \to \mathbf{Set}_*$.*

Proof. Let $\lambda: L \to_* L'$ be a partial function between sets of labels and

$T = (S, i, L, Tran)$ a labelled transition system. Let $\lambda_!(T) = (S', i', L', Tran')$ and $f = (q, \lambda)$ as defined above. In order to show it is cocartesian let

$$\lambda'' : L \to_* L'' \qquad \text{and} \qquad \lambda' : L' \to_* L''$$

be partial functions such that

$$\lambda' \circ \lambda = \lambda'',$$

and $f'' : T \to T''$ be a morphism of labelled transition systems such that $p(f'') = \lambda''$. We require the existence of a unique morphism $f' : \lambda_!(T) \to T''$ such that $p(f') = \lambda'$ and $f' \circ f = f''$.

The morphism f'' must have the form (σ, λ''). Because $\lambda' \circ \lambda = \lambda''$, for any label α if $\lambda(\alpha)$ is undefined so is $\lambda''(\alpha)$. As (σ, λ'') is a morphism this entails that

$$s \sim t \Rightarrow \sigma(s) = \sigma(t)$$

for all states s, t of T. Hence σ factors as

$$\sigma = \sigma' \circ q$$

for a unique $\sigma' : S' \to S''$. Take $f' = (\sigma', \lambda')$. Now, provided f' is a morphism $\lambda_!(T) \to T''$, it is the unique one such that $f'f = f''$. However, $\sigma'(i') = \sigma' q(i) = \sigma(i) = i''$. Any transition of $\lambda_!(T)$ has the form $(q(s), \lambda(\alpha), q(t))$, where (s, α, t) is a transition of T and $\lambda(\alpha)$ is defined. We then see

$$(\sigma' q(s), \lambda'\lambda(\alpha), \sigma' q(t)) = (\sigma(s), \lambda''(\alpha), \sigma(t)),$$

which is a transition of T'' because $f'' = (\sigma, \lambda'')$ is a morphism. Hence f' is a morphism. It follows that f is cocartesian, as required. ▌

As a corollary we obtain:

2.13. THEOREM. *The projection $p : \mathbf{T} \to \mathbf{Set}_*$ determines a cofibration.*

As we have seen, the fibration p with an explicit choice of cartesian liftings yields a functor $\lambda^* : p^{-1}(L') \to p^{-1}(L)$ for any $\lambda : L \to_* L'$. Recall that in general such an explicit choice is given by a cleavage for the fibration. Similarly, the cofibration yields a functor making use of the explicit cocartesian liftings $d(T, \lambda) = (q, \lambda) : T \to \lambda_!(T)$ given in 2.12. In general, a function like d providing a choice of cocartesian liftings will be called a *cocleavage* of the associated cofibration. For any $\lambda : L \to_* L'$, the cofibration p and its cocleavage d determine a functor $\lambda_! : p^{-1}(L) \to p^{-1}(L')$ which acts on morphisms as follows:

There are cocartesian morphisms

$$d(T_0, \lambda)\colon T_0 \to \lambda_!(T_0) \qquad \text{and} \qquad d(T_1, \lambda)\colon T_1 \to \lambda_!(T_1).$$

As $d(T_0, \lambda)$ is cocartesian, any morphism $f\colon T_0 \to T_1$ in $p^{-1}(L)$ determines a unique morphism $f'\colon \lambda_!(T_0) \to \lambda_!(T_1)$ in $p^{-1}(L')$ such that $d(T_1, \lambda) \circ f = f' \circ d(T_0, \lambda)$. This morphism f' we take as the value of $\lambda_!(f)$.

The construction above is quite general and shows how to construct functors between fibres from a cocleavage of an arbitrary cofibration.

It follows from the next result that the two functors $\lambda_!$ and λ^* between fibres, arising from a morphism λ in $\mathbf{Set}_*$, are adjoint to each other. The result is established in general, for an arbitrary functor which is simultaneously a fibration and a cofibration.

2.14. LEMMA. *Suppose $p\colon \mathbf{X} \to \mathbf{B}$ is a fibration and a cofibration. Let $\lambda\colon A \to B$ be a morphism in $\mathbf{B}$. Functors $\lambda_!\colon p^{-1}(A) \to p^{-1}(B)$ determined by a cocleavage for p and $\lambda^*\colon p^{-1}(B) \to p^{-1}(A)$ determined by a cleavage for p form an adjunction between fibres in which $\lambda_!$ is left adjoint to λ^*.*

Proof. Assume $p\colon \mathbf{X} \to \mathbf{B}$ forms a fibration and cofibration. Let $\lambda\colon A \to B$ be a morphism in the base category $\mathbf{B}$.

Let $X \in p^{-1}(A)$ and $Y \in p^{-1}(B)$. Write $r\colon \lambda^*(Y) \to Y$ for the cartesian lifting of λ with respect to Y given by the cleavage for the fibration. Write $l\colon X \to \lambda_!(X)$ for the cocartesian lifting of λ with respect to X given by the cocleavage for the cofibration.

Given a vertical morphism $f\colon \lambda_!(X) \to Y$, the fact r is cartesian entails the existence of a unique vertical morphism $\phi(f)\colon X \to \lambda^*(Y)$ such that

$$r \circ \phi(f) = f \circ l.$$

The additional fact that l is cocartesian ensures ϕ is a bijection from morphisms $\lambda_!(X) \to Y$ in $p^{-1}(B)$ to morphisms $X \to \lambda^*(Y)$ in $p^{-1}(A)$.

To show $\lambda_!$ is left adjoint to λ^* we require that the bijection ϕ satisfy the naturality conditions (MacLane, 1971, p. 79)

$$\text{(i)} \quad \phi(k \circ f) = \lambda^*(k) \circ \phi(f), \qquad \text{(ii)} \quad \phi(f \circ \lambda_!(h)) = \phi(f) \circ h,$$

for all $f\colon \lambda_!(X) \to Y$, $k\colon Y \to Y'$ in $p^{-1}(B)$, and $h\colon X' \to X$ in $p^{-1}(A)$.

In showing (i), let $r'\colon \lambda^*(Y') \to Y'$ be the cartesian lifting of λ given by the cleavage. From the definition of the functor λ^* we see

$$r' \circ \lambda^*(k) = k \circ r.$$

From the definition of ϕ, we see

$$r \circ \phi(f) = f \circ l.$$

Hence

$$r' \circ (\lambda^*(k) \circ \phi(f)) = k \circ r \circ \phi(f) = k \circ f \circ l.$$

Furthermore, $\lambda^*(k) \circ \phi(f)$ is vertical because both $\lambda^*(k)$ and $\phi(f)$ are. But $\phi(k \circ f)$ is defined to be the unique vertical morphism such that

$$r' \circ \phi(k \circ f) = k \circ f \circ l.$$

Hence $\phi(k \circ f) = \lambda^*(k) \circ \phi(f)$, so fulfilling (i).

The argument for (ii is analogous. Let $l' : X' \to \lambda_!(X')$ denote the cocartesian lifting of λ given by the cocleavage. From the definition of $\lambda_!$ as a functor, we obtain

$$\lambda_!(h) \circ l' = l \circ h.$$

From the definition of ϕ, we have

$$r \circ \phi(f) = f \circ l.$$

Combining these facts we obtain

$$r \circ (\phi(f) \circ h) = f \circ l \circ h = (f \circ \lambda_!(h)) \circ l'.$$

Clearly $\phi(f) \circ h$ is vertical. The definition of ϕ characterising $\phi(f \circ \lambda_!(h))$ as the unique vertical morphism g such that $r \circ g = (f \circ \lambda_!(h)) \circ l'$ implies the naturality condition (ii).

We conclude that $\lambda_!$ is left adjoint to λ^*. ▌

The operations on labelled transition systems associated with $\lambda_!$ and λ^* are at least familiar for special kinds of function λ. More curious from the viewpoint of parallel computation is the fact that when λ is total there is a right adjoint to λ^*. defined as follows:

2.15. Definition. Let $\lambda : L \to_* L'$ be total between labelling sets. Let $T = (S, i, L, Tran)$ be a labelled transition system. Define $\lambda_*(T) = (S, i, L', Tran')$, where

$$(s, \beta, t) \in Tran' \Leftrightarrow \forall \alpha . \beta = \lambda(\alpha) \Rightarrow (s, \alpha, t) \in Tran.$$

We sum up the relation between the various functors $\lambda_!$, λ^*, λ_* in:

2.16. Theorem. *For $\lambda : L \to_* L'$, the functors $\lambda_!$ and λ^* form an adjunction from $p^{-1}(L)$ to $p^{-1}(L')$, with $\lambda_!$ the left and λ^* the right adjoint. The functor λ^* has a right adjoint iff λ is total; if λ is total λ_* is right adjoint to λ^*.*

Proof. The fact that $\lambda_!$ is left adjoint to λ^* follows as a special case of Lemma 2.14. It is easy to show λ^* cannot have a right adjoint when λ is truly partial; then λ^* does not preserve initial objects as it would were this so. Suppose $\lambda: L \rightarrow_* L'$ is undefined on label α. Then the transition system $nil_L = (\{i\}, i, L, \varnothing)$ and $nil_{L'} = (\{i\}, i, L', \varnothing)$ are initial objects in the fibres $p^{-1}(L)$ and $p^{-1}(L')$. Certainly nil_L has no transitions. However, $\lambda^*(nil_{L'})$ contains at least one, the transition (i, α, i). To see why λ_* is right adjoint to λ^* when λ is total, let $T \in p^{-1}(L)$ and $T' \in p^{-1}(L')$, and observe that

$$(\sigma, 1_L): \lambda^*(T') \rightarrow T \text{ is a morphism } \Leftrightarrow (\sigma, 1_{L'}): T' \rightarrow \lambda_*(T) \text{ is a morphism.}$$

The correspondence between morphisms $\lambda^*(T') \rightarrow T$ and $T' \rightarrow \lambda_*(T)$ checks out to be a natural isomorphism, making λ^* left adjoint to λ_*. ∎

Thus the functors λ^* and $\lambda_!$ are in a pleasing relation with each other. The construction of the right adjoint λ_* in the case when λ is total is curious in the context of transition systems, but I can see no direct use for it.[2]

3. Constructions on Labelled Transition Systems

The category of labelled transition systems is rich in categorical constructions which furnish the basic combinators for languages of parallel processes. I shall not always give proofs of the universal properties; this is because they are straightforward and full proofs of analogous results for unlabelled transition systems appear in Winskel (1985b).

3.1. *Simple Constructions*

The *nil* transition system

$$nil = (\{i\}, i, \varnothing, \varnothing),$$

the transition system consisting of a single initial state b, is initial and terminal in the category of labelled transition systems.

For a set L of labels,

$$nil_L = (\{i\}, i, L, \varnothing)$$

is initial in the fibre $p^{-1}(L)$ and in its subcategory of safe transition systems. Note $nil = nil_\varnothing$.

[2] A property one might expect of the pair of adjoints $\lambda_!$ and λ^* is the Beck–Chevalley property. We remark that the Beck–Chevalley property is satisfied for all pullback squares in the base category of p, though the fact that the Beck–Chevalley property holds depends crucially on the fact that labelled transition systems are extensional, i.e., for a label α there is at most one α transition between any pair of states.

For a set of labels L, the labelled transition system

$$null_L = (\{i\}, i, L, \{(i, \alpha, i) \mid \alpha \in L\}),$$

consisting of looping α transitions at the initial state for each label α, is terminal in the fibre $p^{-1}(L)$. Note $nil = null_{\varnothing}$.

3.2. *The Product of Labelled Transition Systems*

The product in the category of labelled transition systems is central to representing on transition systems the parallel compositions of processes of languages, which like CCS, CSP, and Occam, communicate by events of synchronisation.

3.2.1. Definition. Assume labelled transition systems $T_0 = (S_0, i_0, L_0, Tran_0)$ and $T_1 = (S_1, i_1, L_1, Tran_1)$. Their product $T_0 \times T_1$ is $(S, i, L, Tran)$, where

$$S = S_0 \times S_1, \qquad \text{with} \quad i = (i_0, i_1),$$

and projections $\rho_0 \colon S_0 \times S_1 \to S_0$, $\rho_1 \colon S_0 \times S_1 \to S_1$

$$\begin{aligned} L = L_0 \times_* L_1 = {} & \{(\alpha, *) \mid \alpha \in L_0\} \cup \{(*, \beta) \mid \beta \in L_1\} \\ & \cup \{(\alpha, \beta) \mid \alpha \in L_0, \beta \in L_1\}, \end{aligned}$$

with projections π_0, π_1, and

$$\begin{aligned} (s, \alpha, s') \in Tran_* \Leftrightarrow {} & (\rho_0(s), \pi_0(\alpha), \rho_0(s')) \in Tran_{0_*} \\ & \& \ (\rho_1(s), \pi_1(\alpha), \rho_1(s')) \in Tran_{1_*}. \end{aligned}$$

Define

$$\Pi_0 = (\rho_0, \pi_0) \qquad \text{and} \qquad \Pi_1 = (\rho_1, \pi_1).$$

Intuitively, transitions with labels of the form (α, β) represent synchronisations between two processes set in parallel, while those labelled $(\alpha, *)$ or $(*, \beta)$ involve only one process, performing transitions unsynchronised with the other. Clearly, this is far too generous a parallel composition to be useful as it stands, allowing as it does all possible synchronisations and absences of synchronisations between two processes. However, as we discuss in Section 3.4, familiar and useful parallel compositions can be obtained from the product operation by further applications of restriction (to remove unwanted synchronisations and perhaps disallow their absences) and relabelling (to rename the results of synchronisations).

3.2.2. Proposition. *Let T_0 and T_1 be labelled transition systems. The construction $T_0 \times T_1$, above, is a product in the category* **T**, *with projections $\Pi_0 = (\rho_0, \pi_0)$, $\Pi_1 = (\rho_1, \pi_1)$. A state s is reachable in $T_0 \times T_1$ iff $\rho_0(s)$ is reachable in T_0 and $\rho_1(s)$ is reachable in T_1.*

Although we have considered only binary products, all products exist in the category of labelled transition systems.

3.3. *The Fibre Product of Transition Systems*

In particular it is worth paying attention to the product in a fibre. It is closely related to the kind of construction one sees on transition systems to model parallel compositions like those in path expressions and in "theoretical CSP" where synchronisations occur between actions of the same nature, as specified by the labelling (see Campbell and Habermann, 1973; Hoare *et al.*, 1984; Lauer and Campbell, 1974). General products and arbitrary parallel compositions can be defined in terms of it with the help of cartesian liftings.

3.3.1. Definition. Let $T_0 = (S_0, i_0, L, Tran_0)$ and $T_1 = (S_1, i_1, L, Tran_1)$ be labelled transition systems over a common set of labels L. Their *fibred product* is $T_0 \times_L T_1 = (S, i, L, Tran)$, where: Its states have the form $S = S_0 \times S_1$, the cartesian product of S_0 and S_1 with projections $\rho_0 : S_0 \times S_1 \to S_0$ and $\rho_1 : S_0 \times S_1 \to S_1$. Take $i = (i_0, i_1)$ as the initial state of the product. Its set of transitions $Tran$ is obtained by taking

$$(s, \alpha, t) \in Tran \Leftrightarrow (\rho_0(s), \alpha, \rho_0(t)) \in Tran_0 \quad \& \quad (\rho_1(s), \alpha, \rho_1(t)) \in Tran_1.$$

3.3.2. Proposition. *Let T_0 and T_1 be labelled transition systems with the same labelled set L. Their fibre product $T_0 \times_L T_1$ is a product in the fibre $p^{-1}(L)$ with projections $(\rho_0, 1_L)$, $(\rho_1, 1_L)$ using the same notation as in the definition above.*

The product of labelled transition systems over different labelling sets L_0 and L_1 can be derived from the product in the fibre over a product of labels using the functors λ_0^* and λ_1^* obtained from the projections $\lambda_0 : L_0 \times_* L_1 \to_* L_0$ and $\lambda_1^* : L_0 \times_* L_1 \to_* L_1$ from the product of labelling sets. This follows from a general fact. Say a category has I-products, for a set I, if it has all products of size I.

3.3.3. Lemma. *Assume $p : X \to B$ is a fibration. Assume the base category* **B** *and the fibres $p^{-1}(B)$, for B in* **B**, *have I-products. Assume functors $\lambda^* : p^{-1}(B') \to p^{-1}(B)$, for $\lambda : B \to B'$ in* **B**, *determined by a cleavage for p, preserve products. Then the category* **X** *has I-products given in the following way:*

Let $X_i \in p^{-1}(B_i)$ for $i \in I$. Let B, $\lambda_i \colon B \to B_i$ for $i \in I$, be a product in **B**. *Let $c_i \colon \lambda_i^*(X_i) \to X_i$ be the cartesian liftings of λ_i given by the cleavage. Let X, q_i, where $i \in I$, be a product of the objects $\lambda_i^*(X_i)$, where $i \in I$, in the fibre $p^{-1}(B)$. Then X, $c_i \circ q_i$ for $i \in I$, is a product in* **X**.

Proof. Under the assumptions stated, we prove that X with projections $c_i \circ q_i$, for $i \in I$, is a product.

Suppose $X' \in p^{-1}(B')$ and $f_i \colon X \to X_i$ are morphisms in **X** for all $i \in I$. Projecting to the base category we obtain a family of morphisms $p(f_i) \colon B' \to B_i$, for $i \in I$, in **B**. As B, λ_i, where $i \in I$, is a product in **B** there is a unique $\lambda \colon B' \to B$ such that

$$p(f_i) = \lambda_i \circ \lambda$$

for all $i \in I$.

Associated with λ^* are the cartesian liftings

$$d_i \colon \lambda^* \lambda_i^*(X_i) \to \lambda_i^*(X_i) \qquad \text{for} \quad i \in I$$

and

$$d \colon \lambda^*(X) \to X$$

specified by the cleavage of the fibration. For all $i \in I$, the definition of how λ^* acts on morphisms gives

$$d_i \circ \lambda^*(q_i) = q_i \circ d. \tag{1}$$

For each $i \in I$ the composition $c_i \circ d_i \colon \lambda^* \lambda_i(X_i) \to X_i$ of cartesian morphisms is itself cartesian. Hence for each i the morphism f_i factors as

$$c_i \circ d_i \circ f_i' = f_i \tag{2}$$

for a unique vertical morphism $f_i' \colon X' \to \lambda^* \lambda_i^*(X_i)$. But, by assumption, $\lambda^*(X)$ with projections $\lambda^*(q_i)$, for $i \in I$, is a product in $p^{-1}(B')$. Thus there is a unique morphism $f' \colon X' \to \lambda^*(X)$ in $p^{-1}(B')$ such that

$$\lambda^*(q_i) \circ f' = f_i' \tag{3}$$

for all $i \in I$. Hence we obtain a morphism $f = d \circ f' \colon X' \to X$ for which

$$\begin{aligned} c_i \circ q_i \circ f &= c_i \circ q_i \circ f' && \text{by definition} \\ &= c_i \circ d_i \circ \lambda^*(q_i) \circ f' && \text{by (1)} \\ &= c_i \circ d_i \circ f_i' && \text{by (3)} \\ &= f_i && \text{by (2)} \end{aligned}$$

for all $i \in I$.

Indeed morphisms $g: X' \to X$ are uniquely determined by the property that

$$c_i \circ q_i \circ g = f_i \qquad \text{for all} \quad i \in I. \tag{4}$$

Assume g satisfies property (4). As d is cartesian, g factors as

$$g = d \circ g' \tag{5}$$

for a vertical $g': X' \to \lambda^*(X)$. Now, for each $i \in I$,

$$\begin{aligned} c_i \circ d_i \circ (\lambda^*(q_i) \circ g') &= c_i \circ q_i \circ d \circ g' && \text{by (1)} \\ &= c_i \circ q_i \circ g && \text{by (5)} \\ &= f_i. \end{aligned}$$

But recall (2), which says: for each i

$$c_i \circ d_i \circ f_i' = f_i \tag{2}$$

for a unique vertical morphism $f_i': X' \to \lambda^* \lambda_i^*(X_i)$. Hence $\lambda^*(q_i) \circ g' = f_i'$ for all $i \in I$. The fact that $\lambda^*(X)$, $\lambda^*(q_i)$ for $i \in I$, is a product in $p^{-1}(B')$ ensures $g' = f'$ and so that $g = f$.

We conclude that X, with projections $c_i \circ q_i$ for $i \in I$, is a product in **X**. This case is typical and shows that **X** has products generally. ▌

In particular we see how to obtain products from fibre products:

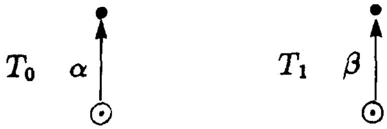
3.3.4. Proposition. *Let T_0 and T_1 be labelled transition systems with sets of labels L_0, L_1, respectively. Let $L_0 \times_* L_1, \lambda_0, \lambda_1$ be the product of their labelling sets. There is an isomorphism*

$$T_0 \times T_1 \cong \lambda_0^*(T_0) \times_{L_0 \times_* L_1} \lambda_1^*(T_1).$$

Proof. Because each λ^* is a right adjoint it preserves products. The proposition now follows from Lemma 3.3.3. ▌

So, not only is the construction $T_0 \times T_1$ a product but, for general reasons, it coincides with the fibre product of the transition systems $\lambda_0^*(T_0)$ and $\lambda_1^*(T_1)$, brought into a common fibre via cartesian liftings of the projections λ_0, λ_1 on labelling sets. We illustrate this on an example.

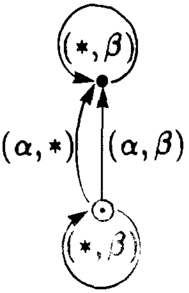
3.3.5. Example. Let T_0 and T_1 be the labelled transition systems



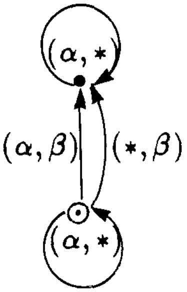
where T_0 has $\{\alpha\}$ and T_1 has $\{\beta\}$ as labelling set. The product of these labelling sets is

$$\{\alpha\} \times_* \{\beta\} = \{(\alpha, *), (\alpha, \beta), (*, \beta)\}$$

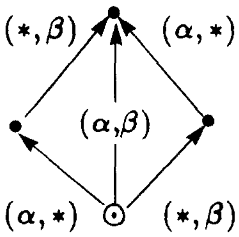
with projections λ_0 onto the first coordinate and λ_1 onto the second. Thus $\lambda_0(\alpha, *) = \lambda_0(\alpha, \beta) = \alpha$ and $\lambda_0(*, \beta) = *$. The transition system $\lambda_0^*(T_0)$ has the form



containing a $(*, \beta)$-stutter, introduced through the undefinedness of λ_0 on $(*, \beta)$. Similarly $\lambda_1^*(T_1)$ has the form



this time with a $(*, \alpha)$-stutter. Because of the existence of the stutters, for the fibre product $\lambda_0^* (T_0) \times_{\{\alpha\} \times_* \{\beta\}} \lambda_1^*(T_1)$ we obtain, as is to be expected from 3.3.4, the product



3.4. *Parallel Compositions.*

In the present framework, we do not obtain arbitrary parallel compositions as single universal constructions. Instead, they come about as a result of first taking a product of T_0 and T_1, with labelling sets L_0, L_1 respectively, to give a transition system $T_0 \times T_1$ with labelling set $L_0 \times_* L_1$, then restricting by taking $i^*(T_0 \times T_1)$ for an inclusion $i: S \to L_0 \times_* L_1$, followed by a relabelling $r_!(i^*(T_0 \times T_1))$ with respect to a total $r: S \to_* L$. In this way, using a combination of product, restriction, and relabelling we can represent all conceivable parallel compositions which occur by synchronisation.

In earlier papers such as Winskel (1982, 1985b), it has been shown how parallel compositions based on synchronisation can be viewed, in a uniform way, as arising from *synchronisation algebras*. A synchronisation algebra on a set L of labels (not containing the distinct elements $*$, 0) consists of a binary, commutative, associative operation $\cdot$ on $L \cup \{*, 0\}$ such that

$$\alpha \cdot 0 = 0 \qquad \text{and} \qquad (\alpha_0 \cdot \alpha_1 = * \Leftrightarrow \alpha_0 = \alpha_1 = *)$$

for all $\alpha, \alpha_0, \alpha_1 \in L \cup \{*, 0\}$. The role of 0 is to specify those synchronisations which are not allowed whereas the composition $\cdot$ specifies a relabelling. For a synchronisation algebra on labels L, let $\lambda_0, \lambda_1: L \times_* L \to_* L$ be the projections on its product in $\mathbf{Set}_*$. The parallel composition of two transition systems T_0, T_1, labelled over L, can be obtained as $r_! i^*(T_0 \times T_1)$, where $i: D \to L \times_* L$ is the inclusion of

$$D = \{a \in L \times_* L \mid \lambda_0(a) \cdot \lambda_1(a) \neq 0\}$$

determined by the 0-element, and $r: D \to L$ is the relabelling, given by

$$r(a) = \lambda_0(a) \cdot \lambda_1(a)$$

for $a \in D$.

The fact that parallel compositions factor into a composition of product, restriction, and relabelling suggests that reasoning about parallel compositions should be factored into three stages, a point of view we adopt later in presenting a compositional proof system.

3.5. *The Coproduct of Labelled Transition Systems*

The category of labelled transition systems has coproducts:

3.5.1. Definition. Let $T_0 = (S_0, i_0, L_0, Tran_0)$ and $T_1 = (S_1, i_1, L_1, Tran_1)$ be labelled transition systems. Their sum $T_0 + T_1$ is $(S, i, L, Tran)$, where

$$S = (S_0 \times \{i_1\}) \cup (\{i_0\} \times S_1) \qquad \text{with } i = (i_0, i_1), \text{ and injections } in_0, in_1$$

$$L = L_0 + L_1 \qquad \text{with injections } j_0, j_1$$

$$t \in Tran \Leftrightarrow \exists (s, \alpha, s') \in Tran_0 . t = (in_0(s), j_0(\alpha), in_0(s'))$$
$$\text{or } \exists (s, \alpha, s') \in Tran_1 . t = (in_1(s), j_1(\alpha), in_1(s')).$$

3.5.2. Proposition. *Let T_0 and T_1 be labelled transition systems. Their sum $T_0 + T_1$, with injections (in_0, j_0), (in_1, j_1), is a coproduct in the category of labelled transition systems. A state s is reachable in a sum iff there is s_0 reachable in T_0 with $s = in_0(s_0)$ or there is s_1 reachable in T_1 with $s = in_1(s_1)$.*

Each fibre $p^{-1}(L)$ has coproducts for a labelling set L. In form they are very similar to sums of labelled transition systems in general—they differ only in the labelling part.

3.5.3. Definition. *The fibre sum of labelled transition systems.* Let $T_0 = (S_0, i_0, L, Tran_0)$ and $T_1 = (S_1, i_1, L, Tran_1)$ be labelled safe transition systems over the same labelling set L. Their fibre sum $T_0 +_L T_1 = (S, i, L, Tran)$ (note that it is over the same labelling set), where

$$S = (S_0 \times \{i_1\}) \cup (\{i_0\} \times S_1) \qquad \text{with} \quad i = (i_0, i_1), \text{ and injections } in_0, in_1,$$

and

$$t \in Tran \Leftrightarrow \exists (s, \alpha, s') \in Tran_0 . t = (in_0(s), \alpha, in_0(s'))$$
$$\text{or } \exists (s, \alpha, s') \in Tran_1 . t = (in_1(s), \alpha, in_1(s')).$$

Fibre sums give coproducts in the fibres for the subcategory of labelled transition systems. This time using the cocartesian liftings of injections on labelling sets we can derive the sum in the category of labelled transition systems in general from the fibre sum—the proof is dual to that provided earlier for products in 3.3.3, 3.3.4.

3.5.4. Proposition. *Let T_0 and T_1 be labelled transition systems over L. The labelled transition system $T_0 +_L T_1$ with injections $(\iota_0, 1_L)$ and $(\iota_1, 1_L)$, as defined above, is a coproduct in the subcategory of labelled transition systems over L.*

Let T_0 and T_1 be labelled transition systems over L_0 and L_1, respectively. Let $L_0 + L_1$ have injections $j_k : L_k \to L_0 + L_1$, for $k = 0, 1$. Then

$$T_0 + T_1 \cong j_{0!} T_0 +_{(L_0 + L_1)} j_{1!} T_1 .$$

Only coproducts of two labelled transition systems have been considered. All coproducts exist in fibres and in the category of all labelled transition systems. Thus there are indexed sums of labelled transition systems of the kind used in CCS. As is pointed out by Winskel (1985b), the sum construction on transition systems is of the form required for CCS when the transition systems are "nonrestarting," i.e., have no transitions back to the initial state.

The categorical constructions form a basis for languages of parallel processes with constructs like parallel compositions and nondeterministic sums. The cartesian and cocartesian liftings give rise to restriction and relabelling operations as special cases, but the more general constructions, arising for morphisms in the base category which are truly partial, might also be useful constructions to introduce into a programming language. For instance, $\lambda_!$ might be a convenient looping construction—the idea is illustrated in Example 2.11—though identifying two states by such a looping construction depends on first having a transition between them, and it is not clear how best to do this.

This raises an omission from our collection of constructions; we have not yet mentioned an operation which introduces new transitions from scratch. Traditionally, in languages like CCS, CSP, and Occam this is done with some form of prefixing operation, the effect of which is to produce a new process which behaves like a given process once a specified, initial action has taken place. While there is no difficulty in defining a prefixing operation, even though it is functorial, it is not clear what its categorical status should be (and surely in a more complete presentation of categorical models for parallelism it will have a more prominent role than it does at present). Prefixing, formally defined here, will play a role later in Section 6.

3.5.5. Definition. Let $T=(S, i, L, Tran)$ be a labelled transition system. Let α be a label (not $*$). Define the prefix $\alpha T=(S', i', L', Tran')$, where

$$\begin{aligned} S' &= \{\{s\} \mid s \in S\} \cup \{\varnothing\}, \\ i' &= \varnothing, \\ L' &= L \cup \{\alpha\}, \\ Tran' &= Tran \cup \{(\varnothing, \alpha, \{i\})\}. \end{aligned}$$

4. Synchronisation Trees

In his foundational work on CCS, Milner (1980) introduced synchronisation trees as a model of parallel processes and explained the meaning of the language of CCS in terms of operations on them. In this section we briefly examine the category of synchronisation trees and its relation to that of labelled transition systems. This illustrates the method by which many other models are related, and the role category-theoretic ideas play in formulating and proving facts which relate semantics in one model to semantics in another.

A synchronisation tree is a tree together with labels on on its arcs. Formally, we define synchronisation trees to be special kinds of labelled transition systems, those for which the transition relation is acyclic and can only branch away from the root.

4.1. DEFINITION. A *synchronisation tree* is a transition system $(S, i, L, Tran)$, where

(i) every state is reachable,

(ii) the relation $\rightarrow^{+}$ is irreflexive, and

(iii) $s' \rightarrow s \;\&\; s'' \rightarrow s \Rightarrow s' = s''$.

Regarded in this way, we obtain synchronisation trees as a full subcategory of labelled transition systems, with a projection functor to the category of labelling sets with partial functions.

4.2. DEFINITION. Write **S** for the full subcategory of synchronisation trees in **T**.

In fact, the inclusion functor $\mathbf{S} \hookrightarrow \mathbf{T}$ has a right adjoint $\mathcal{U}: \mathbf{T} \rightarrow \mathbf{S}$ which has the effect of unfolding a labelled transition system to a synchronisation tree.

4.3. DEFINITION. Let T be a labelled transition system $(S, i, L, Tran)$. Define $\mathcal{U}(T)$ to be $(S', i', L, Tran')$, where

The set S' consists of all finite, possibly empty, sequences of transitions $(t_1, ..., t_i, t_{i+1}, ..., t_{n-1})$ such that $t_i^{\cdot} = {}^{\cdot}t_{i+1}$ whenever $1 < i < n$.

The element $i' = (\;)$, the empty sequence.

The set $Tran'$ consists of all triples (u, α, v), where $u, v \in S'$ and $v = u((s, \alpha, s'))$, obtained by appending an α transition to u.

Define $\phi: S' \rightarrow S$ by taking $\phi((\;)) = i$ and $\phi((t_1, ..., t_n)) = t_n^{\cdot}$.

4.4. THEOREM. *Let T be a labelled transition system, with labelling set L. Then $\mathcal{U}(T)$ is a synchronisation tree, also with labelling set L, and, with the definition above, $(\phi, 1_L): \mathcal{U}(T) \rightarrow T$ is a morphism. Moreover $\mathcal{U}(T)$, $(\phi, 1_L)$ is cofree over T with respect to the inclusion functor* $\mathbf{S} \hookrightarrow \mathbf{T}$; *i.e., for any* *morphism $f\colon V \rightarrow T$, with V a synchronisation tree, there is a unique morphism $g\colon V \rightarrow \mathscr{U}(T)$ such that $f = (\phi, 1_L)\, g$:*

$$\begin{tikzcd}
T & \mathscr{U}(T) \arrow[l, "(\phi, 1_L)"'] \arrow[d, "g"'] \\
 & V \arrow[ul, "f"]
\end{tikzcd}$$

Proof. Let T be a labelled transition system, with labelling set L. It is easily seen that $\mathscr{U}(T)$ is a labelled transition system with labelling set L and $(\phi, 1_L)\colon \mathscr{U}(T) \rightarrow T$ is a morphism. To show the cofreeness property, let $f = (\sigma, \lambda)\colon V \rightarrow T$ be a morphism from a synchronisation tree V. We require the existence of a unique morphism $g\colon V \rightarrow \mathscr{U}(T)$ such that $f = (\phi, 1_L)\, g$. The morphism g must necessarily have the form $g = (\sigma_1, \lambda)$. The map σ_1 is defined by induction on the distance from the root of states of V, as follows:

On the initial state i_V of V, we take $\sigma_1(i_V) = (\ )$. For any state v' for which (v, α, v') is a transition of V we take $\sigma_1(v') = \sigma(v)$ if $\lambda(\alpha) = *$ and otherwise, in the case where $\lambda(\alpha)$ is defined, take $\sigma_1(v') = \sigma(v)((\sigma(v), \lambda(\alpha), \sigma(v'))$.

It follows by induction on the distance of states v from the root that $\sigma(v) = \phi\sigma_1(v)$, and that (σ_1, λ) is the unique morphism such that $f = (\phi, 1_L)\, g$. (For a very similar, but more detailed, argument, see Winskel, 1985b.) ▌

It follows that the operation $\mathscr{U}$ extends to a functor which is right adjoint to the inclusion functor from **S** to **T** and the morphisms $(\phi, 1_L)\colon \mathscr{U}(T) \rightarrow T$ are the counits of this adjunction (see Maclane, 1971, Theorem 1, p. 81). This makes **S** a coreflective subcategory of **T**, which implies the intuitively obvious fact that a synchronisation tree T is isomorphic to its unfolding $\mathscr{U}(T)$ (see Maclare, 1971, p. 88).

Write $q\colon \mathbf{S} \rightarrow \mathbf{Set}_*$ for the functor obtained as the restriction of the projection functor $p\colon \mathbf{T} \rightarrow \mathbf{Set}_*$. We may reasonably ask whether or not this projection determines a fibration and a cofibration. Fortunately there is no need to go through the rigmarole of constructing the cartesian and cocartesian liftings directly. Cartesian liftings are preserved by the unfolding functor by the following general lemma. So cartesian liftings for synchronisation trees can be produced from cartesian liftings between transition systems ensuring that q is a fibration. The dual of the lemma helps show that q is a cofibration. Referring to the conditions of the lemma, note that the counits of the adjunction between synchronisation trees and labelled transition systems project to identities and so are certainly vertical.

4.5. LEMMA. *Suppose $p: \mathbf{X} \to \mathbf{B}$ and $q: \mathbf{Y} \to \mathbf{B}$ and that functors $L: \mathbf{X} \to \mathbf{Y}$ and $R: \mathbf{Y} \to \mathbf{X}$ form an adjunction with L left adjoint to R in such a way that $qL = p$ and $pR = q$ and each counit $\varepsilon_Y: LR(Y) \to Y$ is vertical for $Y \in \mathbf{Y}$, i.e., $q(\varepsilon_Y) = 1_{q(Y)}$. Then the right adjoint R preserves cartesian morphisms.*

Proof. Suppose $f: Y \to Y'$ is cartesian and $g(f) = \lambda: B \to B'$. We need that $R(f): R(Y) \to R(Y')$ is cartesian over $p(R(f)) = \lambda$. Let $g': X \to R(Y')$ be such that $p(g') = \lambda': B'' \to B'$ and $\lambda' = \lambda \circ \lambda''$ for $\lambda'': B'' \to B$. We are required to show

$$\exists! g'': X \to R(Y).p(g'') = \lambda'' \qquad \& \qquad g' = (R(f)) \circ g''. \qquad (*)$$

Let

$$\varepsilon: LR(Y) \to Y \qquad \text{and} \qquad \varepsilon': LR(Y') \to Y'$$

be counits of the adjunction. Consider the morphism $\varepsilon' \circ (L(g')): L(X) \to Y'$. We have

$$q(\varepsilon' \circ (L(g'))) = q(\varepsilon') \circ q(L(g')) = 1'_B \circ p(g') = \lambda'.$$

As $f: Y \to Y'$ is cartesian there is a unique $k: L(X) \to Y$ with $q(k) = \lambda''$ and

$$f \circ k = \varepsilon' \circ (L(g')).$$

Now by the cofreeness of $\varepsilon: LR(Y) \to Y$, there is a unique $g'': X \to R(Y)$ such that

$$\varepsilon \circ (L(g'')) = k.$$

Also,

$$p(g'') = qL(g'') = 1_B \circ (qL(g'')) = q(\varepsilon \circ (L(g''))) = q(k) = \lambda''.$$

From the naturality of the adjunction we have

$$\varepsilon' \circ (LR(f)) = f \circ \varepsilon.$$

Using this fact we obtain

$$\begin{aligned} \varepsilon \circ L((R(f)) \circ g'') &= \varepsilon' \circ (LR(f)) \circ (L(g'')) \\ &= f \circ \varepsilon \circ (L(g'')) \\ &= f \circ k \\ &= \varepsilon' \circ (L(g')). \end{aligned}$$

But $\varepsilon' : LR(Y)' \to Y'$ is cofree, so by the accompanying uniqueness property we obtain

$$g' = (R(f)) \circ g''.$$

Thus we have fulfilled the existence part of the requirement $(*)$.

To show uniqueness, assume also that

$$g_1 : X \to R(Y) \qquad \& \qquad p(g_1) = \lambda'' \qquad \& \qquad g' = (R(f)) \circ g_1.$$

Then

$$\begin{aligned} \varepsilon' \circ (L(g')) &= \varepsilon' \circ L(R(f) \circ g_1) \\ &= \varepsilon' \circ (LR(f)) \circ (L(g_1)) \\ &= f \circ \varepsilon \circ (L(g_1)) \qquad \text{by naturality of the adjunction.} \end{aligned}$$

Recall

$$\varepsilon' \circ (L(g')) = f \circ k = f \circ \varepsilon \circ (L(g'')).$$

Thus

$$f \circ (\varepsilon \circ (L(g_1))) = f \circ (\varepsilon \circ (L(g''))).$$

But f is cartesian so $\varepsilon \circ (L(g_1)) = \varepsilon \circ (L(g''))$. Finally by the cofreeness of ε we obtain $g_1 = g''$, as required for the uniqueness part of $(*)$. ▌

The property that the counits are vertical could be replaced equivalently by one saying the units are vertical, or by one saying that the adjunction cuts down to adjunctions between fibres over common objects in the base category.

As a corollary, we obtain:

4.6. Corollary. *The functor* $q : \mathbf{S} \to \mathbf{Set}_*$ *forms a fibration and a cofibration.*

Proof. Let $\lambda : L \to L'$ be a morphism in the base category $\mathbf{Set}_*$. Let S be a synchronisation tree with labelling set L'. From the fact that $p : \mathbf{T} \to \mathbf{Set}_*$ forms a fibration there is a cartesian morphism $f : T \to S$ such that $p(f) = \lambda$. By Lemma 4.5, $\mathscr{U}(f) : \mathscr{U}(T) \to \mathscr{U}(S)$ is cartesian with respect to $q : \mathbf{S} \to \mathbf{Set}_*$. But the counit $\varepsilon : \mathscr{U}(S) \cong S$ of the coreflection is vertical, providing a cartesian morphism $\varepsilon \circ \mathscr{U}(f) : \mathscr{U}(T) \to S$ for which $q(\varepsilon \circ \mathscr{U}(f)) = \lambda$. Hence $q : \mathbf{S} \to \mathbf{Set}_*$ forms a fibration.

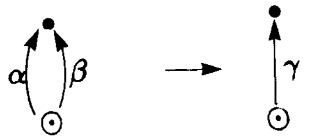
Again, let $\lambda : L \to L'$ be a morphism in the base category $\mathbf{Set}_*$. First, observe that if S is a synchronisation tree then so is $\lambda_!(S)$. By the dual of

Lemma 4.5, the inclusion functor $\mathbf{S} \hookrightarrow \mathbf{T}$ preserves cocartesian morphisms when they exist. It follows that when S is a synchronisation tree with labelling set L and $\lambda : L \rightarrow_* L'$, the cocartesian lifting $S \rightarrow \lambda_!(S)$ with respect to $p: \mathbf{T} \rightarrow \mathbf{Set}_*$ is also cocartesian with respect to $q: \mathbf{T} \rightarrow \mathbf{Set}_*$. Hence $q: \mathbf{T} \rightarrow \mathbf{Set}_*$ is a cofibration. ▌

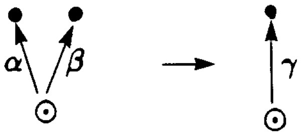
In fact, the proof supplies a cleavage and a cocleavage for q. These determine the functors $\mathscr{U}\lambda^* : q^{-1}(L') \rightarrow q^{-1}(L)$ and $\lambda_! : q^{-1}(L) \rightarrow q^{-1}(L')$ between fibres for a morphism $\lambda : L \rightarrow_* L'$ in the base category—as observed in the proof above $\lambda_!$ restricts to a functor between synchronisation trees. By Lemma 2.14, $\lambda_!$ is left adjoint to $\mathscr{U}\lambda^*$. As is to be expected, when the morphism λ is an inclusion λ^* amounts to the usual restriction operation on synchronisation trees and when total $\lambda_!$ is the usual relabelling.

Note that Lemma 4.5 does not state that the left adjoint L preserves cartesian morphisms. Nor does it entail that the right adjoint R preserves cocartesian morphisms, and these are not true in general. For instance, they do not hold for the coreflection between synchronisation trees and labelled transition systems, as shown in the examples below. On the other hand, $\mathscr{U}$, the unfolding operation, does preserve cocartesian liftings of total functions (although simple to prove directly in this special case, I do not know a general argument which would give the analogous result for the adjunctions relating other models too).

4.7. Example. Let $\lambda : \{\alpha, \beta\} \rightarrow_* \{\gamma\}$ be total such that $\lambda(\alpha) = \lambda(\beta) = \gamma$. The morphism

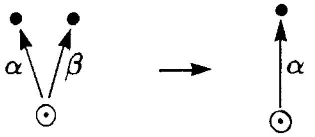


is cartesian with respect to the projection $p: \mathbf{T} \rightarrow \mathbf{Set}_*$ for transition systems. Contrast this with the following cartesian morphism with respect to $q: \mathbf{S} \rightarrow \mathbf{Set}_*$ for synchronisation trees, obtained by unfolding it:

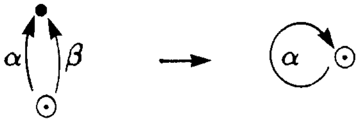


The matter is clearly not cartesian with respect to $p: \mathbf{T} \rightarrow \mathbf{Set}_*$ of transition systems, showing that cartesian morphisms are not preserved by the left adjoint, inclusion functor from synchronisation trees to labelled transition systems.

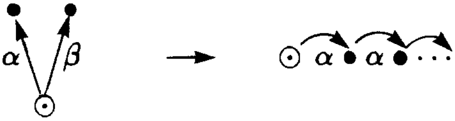
Now, let $\lambda : \{\alpha, \beta\} \to_* \{\alpha\}$ be such that $\lambda(\alpha) = \alpha$ and $\lambda(\beta)$ is undefined. The morphism



is cocartesian with respect to the projection $p : \mathbf{T} \to \mathbf{Set}_*$ and so also with respect to $q : \mathbf{S} \to \mathbf{Set}_*$. With respect to p, the morphism



is cocartesian. However, its image under the unfolding operation $\mathscr{U}$ is



which cannot be cocartesian with respect to q, given the earlier form of the cocartesian lifting of λ with respect to the same object.

Like labelled transition systems, synchronisation trees have been used to give semantics to languages like CCS and CSP (see, e.g., Milner, 1980; Brookes, 1983). Nondeterministic sums of processes are modelled by the operation of joining synchronisation trees at their roots, a coproduct of synchronisation trees and a special case of the coproduct of transition systems. We use $\sum_{i \in I} S_i$ for the sum of synchronisation trees indexed by $i \in I$. For the semantics of parallel composition, use is generally made of Milner's "expansion theorem" (see Milner, 1980). In our context, the expansion of parallel composition as a nondeterministic sum appears as a characterisation of the product of synchronisation trees. The product of two synchronisation trees S and T of the form

$$S \cong \sum_{i \in I} \alpha_i S_i \qquad \text{and} \qquad T \cong \sum_{j \in J} \beta_j T_j$$

is given by

$$S \times T \cong \sum_{i \in I} (\alpha_i, *)\, S_i \times T + \sum_{i \in I, j \in J} (\alpha_i, \beta_j)\, S_i \times T_j + \sum_{j \in J} (*, \beta_j)\, S \times T_j.$$

(See Winskel (1985b, 1987b) for a very similar result, with proof.) The fact that the category of synchronisation trees has products and that they are preserved by the unfolding operation $\mathscr{U}$ is a consequence of the general fact that right adjoints preserve limits.

This indicates how the coreflection between synchronisation trees and labelled transition systems and the fact that they form fibrations and cofibrations help in formulating and proving the relationship between different models. For labelled transition systems and synchronisation trees general facts like the existence of an adjunction or the preservation of cartesian morphisms by functors can be brought to bear on proofs showing, for instance, how semantics is preserved in passing from one model to another. Such techniques are available only by virtue of placing models for parallelism in a categorical setting. Although we are far from a complete understanding of models for parallel processes, it is fortunate that the category-theoretic view of parallel processes, illustrated here for labelled transition systems and synchronisation trees, works for a variety of models.

We could hardly expect the categories of labelled transition systems and synchronisation trees, their properties and their relationship, to be more straightforward. When it comes to other models of parallel computation the situation can be less ideal. For example, although there is a coreflection from a category of labelled event structures to a category of safe Petri nets, neither category forms a fibration with the definitions of these structures as they are usually given. Still enough cartesian and cocartesian maps do exist in these other categories to model restriction and relabelling. When they exist, cartesian liftings will be preserved by the right adjoints in the adjunctions between models. This is a consequence of Lemma 4.5. In fact, all the adjunctions established by (Winskel (1987b) generalise to labelled structures in such a way as to satisfy the property required by Lemma 4.5. On the other hand, because we do not obtain fibrations from all of these categories, there are not necessarily analogues of Theorem 3.3.4, showing how the product can be obtained from the product in the fibre.[3]

5. The Preservation and Reflection of Properties by Morphisms

We now begin a new line, that of putting a logic on the category of labelled transition systems. We start by examining a simple modal logic to express properties of labelled transition systems.

[3] Not too much emphasis should be placed on the fact that models like labelled safe Petri nets and labelled event structures do not form fibrations; even they can be made into fibrations by breaking with tradition and changing their definitions slightly. By permitting more objects in the categories, by, for example, allowing safe nets to have *independent* events with no pre- or postconditions, one obtains fibrations; the role of an α stutter in transition systems is played by α-labelled independent events in nets. See Winskel (1988) for a discussion.

Assume $T = (S, i, L, Tran)$ is a labelled transition system. We associate with it a language of modal assertions. It has atomic assertions corresponding to true and false, an assertion true just at the initial state, as well as other basic assertions whose interpretation we leave open. More complicated assertions are formed by using propositional connectives, two modal operators, and recursively defined assertions, using a least fixed point operator, which make use of assertion variables assumed to form a set Var. As assertions for T we take

$$\phi ::= \mathit{basic} \mid I \mid \mathrm{tt} \mid \mathrm{ff} \mid \phi_0 \wedge \phi_1 \mid \phi_0 \vee \phi_1 \mid \neg\phi \mid \langle a\rangle \phi \mid \phi\langle a\rangle \mid X \mid \mu X.\phi,$$

where *basic* ranges over atomic assertions, a is a label in L or $*$, and X is an assertion variable from Var. There are the usual notions of free and bound occurrence of a variable, and, as usual, we impose a restriction on occurrences within the scope of X in assertions $\mu X.\phi$; they should be under an even number of negation signs. We write $\phi\colon T$ to indicate ϕ is an assertion of T. We shall interpret the logic classically and so can define implication $\phi_0 \rightarrow \phi_1$ and logical equivalence $\phi_0 \leftrightarrow \phi_1$ in the well-known ways. We use $[a]\phi$ instead of $\neg\langle a\rangle\neg\phi$ and $\phi[a]$ instead of $\neg(\neg\phi\langle a\rangle)$, and as remarked by Kozen (1982), could define a greatest fixed point operator $\nu X.\phi$ by taking it to stand for $\neg\mu X.\neg\phi[\neg X/X]$. Again, as remarked by Kozen (1982), we observe that any assertion ϕ in which all free occurrences of X occur under an even number of negations is logically equivalent to an assertion built up using the additional $[a]$ modalities and considering as additional atoms negations of atomic assertions and perhaps $\neg Y$ for free variables Y *distinct* from X, but with no further use of negation.

As we shall see, an assertion $\phi\colon T$ is satisfied by a subset of reachable states. Such subsets we call properties, and we write $\mathscr{P}(T)$ for the powerset $P(\{s \mid s \text{ is reachable in } T\})$. The assertion I is satisfied only by the initial state. An assertion $\langle a\rangle\phi$ is satisfied by the reachable states from which a can occur and in so doing lead to a state satisfying ϕ. An assertion $\phi\langle a\rangle$ is satisfied by those states which can be reached from a reachable state satisfying ϕ via a similar transition. With such a modality and the help of recursion we can write down $\mu X.I \vee X\langle\alpha\rangle$, an assertion which is satisfied by precisely those states of a transition system which are reachable purely by occurrences of α transitions.

With respect to an environment $\vartheta\colon \mathrm{Var} \rightarrow \mathscr{P}(T)$ for assertion variables, we require

$$[\![I]\!]\,\vartheta = \{i\},\ [\![\mathrm{tt}]\!]\,\vartheta = \{s \mid s \text{ is reachable in } T\},\ [\![\mathrm{ff}]\!]\,\vartheta = \varnothing,$$

$$[\![\phi_0 \wedge \phi_1]\!]\,\vartheta = [\![\phi_0]\!]\,\vartheta \cap [\![\phi_1]\!]\,\vartheta,$$

$$[\![\phi_0 \vee \phi_1]\!]\,\vartheta = [\![\phi_0]\!]\,\vartheta \cup [\![\phi_1]\!]\,\vartheta,$$

$$[\![\neg\phi]\!]\,\vartheta = \{s \mid s \text{ is reachable in } T \,\&\, s \notin [\![\phi]\!]\,\vartheta\},$$

$$[\![\langle a\rangle\,\phi]\!]\vartheta=\{s\,|\,\exists s'\cdot s\xrightarrow{a}s' \text{ in } T\,\&\, s'\in[\![\phi]\!]\vartheta\},$$

$$[\![\phi\langle a\rangle]\!]\vartheta=\{s\,|\,\exists s'\cdot s'\xrightarrow{a}s \text{ in } T\,\&\, s'\in[\![\phi]\!]\vartheta\},$$

$$[\![X]\!]\vartheta=\vartheta(X)$$

$$[\![\mu X.\phi]\!]\,\vartheta=\text{the smallest } S \text{ such that } S=[\![\phi]\!]\;\vartheta[S/X].$$

This leaves the meaning of basic assertions unspecified. Because of the syntactic restriction on ϕ used in recursion, $S\mapsto[\![\phi]\!]\vartheta[S/X]$ is a monotonic operation on $\mathscr{P}(T)$ with respect to inclusion, and so it has a least fixed point by Tarski's theorem.

When ϕ is a closed assertion the denotation $[\![\phi]\!]\vartheta$ is independent of the environment ϑ, and we shall generally write its denotation as just $[\![\phi]\!]$.

A closed assertion ϕ of a transition system T is *valid*, written $T\models\phi$ iff $[\![\phi]\!]=\{s\,|\,s$ is reachable in $T\}$, and so the fact that ϕ is valid means ϕ is invariantly true throughout the reachable states of T. For future reference we present an (incomplete) proof system for transition system assertions, based directly on that of Kozen (1982). The proof rules will be sound in the sense that if ϕ is provable, $\vdash\phi$, for $\phi\colon T$ then $T\models\phi$. Later, we shall incorporate the rules within a proof system based around a term language for transition systems with syntax-directed proof rules. As rules we include those for propositional logic, rules for "farwards" modalities

$$\vdash\langle a\rangle\mathrm{ff}\leftrightarrow\mathrm{ff},\qquad\vdash(\langle a\rangle\phi\vee\langle a\rangle\,\psi)\leftrightarrow\langle a\rangle(\phi\vee\psi),$$

$$\vdash\langle a\rangle\phi\wedge[a]\,\psi\rightarrow\langle a\rangle(\phi\wedge\psi),$$

$$\vdash\langle *\rangle\,\phi\leftrightarrow\phi,$$

similar rules for the "backwards" modalities, and these rules for recursive assertions

$$\vdash\phi[\mu X.\phi/X]\rightarrow\mu X.\phi,\qquad\frac{\vdash\phi[\psi/X]\rightarrow\psi}{\vdash\mu X.\phi\rightarrow\psi}.$$

(See Appendix II for one way to extend this to a full logic.)

We examine assertions on transition systems from an indexed category viewpoint. A morphism $f=(\sigma,\lambda)\colon T\rightarrow T'$ between labelled transition systems determines a function $\mathscr{P}f\colon\mathscr{P}(T')\rightarrow\mathscr{P}(T)$ given by $\mathscr{P}f(V)=\{s\,|\,s$ is reachable in $T\,\&\,\sigma(s)\in V\}$ for $V\in\mathscr{P}(T')$. In this sense each transition system indexes a partial-order category of relevant properties, where the order is inclusion. We can convert the indexing $\mathscr{P}$ into a fibration using a constructon due to Grothendieck.

In general, this recasts a functor $F\colon\mathbf{B}^{\mathrm{op}}\rightarrow\mathbf{Cat}$, indexing categories, as a fibration $p\colon\mathbf{X}\rightarrow\mathbf{B}$ in which the indexed categories reappear as fibres. The objects of $\mathbf{X}$ are pairs $U\colon B$ where $B \in \mathbf{B}$ and $U \in F(B)$. Its morphisms from $U{:}B$ to $U'{:}B'$ are pairs (v,f) where $f{:}B \to B'$ in $\mathbf{B}$ and $v\colon U \to F(U')$ in $F(B)$. Two morphisms $(v,f)\colon (U\colon B) \to (U'\colon B')$ and $(v',f')\colon (U'\colon B') \to (U''\colon B'')$ compose to give $(((Ff)(v'))\circ v),\ f'\circ f)$. This makes $\mathbf{X}$ into a category with identity morphisms $(1_U, 1_B)$ on $U\colon B$ in $\mathbf{X}$. The projection functor $p\colon \mathbf{X} \to \mathbf{B}$ is defined so a morphism $(v,f)\colon (U\colon B) \to (U'\colon B')$ in $\mathbf{X}$ goes to $f\colon B \to B'$ in $\mathbf{B}$. The functor p can be checked to be a fibration; a cartesian lifting of $f\colon B \to B'$ in $\mathbf{B}$ with respect to $U' \in F(B)$ is the morphism $(1_{(Ff)U'}, f)$. Such a fibration p is called the *Grothendieck fibration* of F.

In particular we can construct the Grothendieck fibration $v\colon \mathbf{Prop} \to \mathbf{T}$ of the functor $\mathscr{P}$. The objects of $\mathbf{Prop}$ are pairs $U\colon T$ which associate a property $U \in \mathscr{P}T$ with a labelled transition system T. Morphisms $(U\colon T) \to (U'\colon T')$ are in 1–1 correspondence with morphisms $f\colon T \to T'$ of labelled transition systems such that $U \subseteq (\mathscr{P}F)(U')$. Such morphisms have an intuitive significance, associated with the preservation of properties along morphisms. Assume the component of f between states is the function σ. If $U \subseteq (\mathscr{P}f)(U')$ then whenever s satisfies a property U in T then $\sigma(s)$ satisfies the property U' in T'. We write $U \mathrel{-\!\!-\!\!>}^{f} U'$ to signify this. Another meaningful relation arises slightly differently. If we order properties by reverse inclusion we obtain a functor like $\mathscr{P}$ but taking a labelled transition system T to $(PT)^{\mathrm{op}}$. When we come to construct its Grothendieck fibration the vertical morphisms have changed direction so we can describe it as $v^{\mathrm{vop}}\colon \mathbf{Prop}^{\mathrm{vop}} \to \mathbf{T}$. In more detail, the objects of $\mathbf{Prop}^{\mathrm{vop}}$ are still pairs $U\colon T$ of a labelled transition system and a property, but now the morphisms $(U\colon T) \to (U'\colon T')$ in $\mathbf{Prop}^{\mathrm{vop}}$ correspond to morphisms $f\colon T \to T'$ between labelled transition systems such that $(\mathscr{P}f)(U') \subseteq U$. To see their intuitive significance, note that if $(\mathscr{P}f)(U') \subseteq U$ then whenever $\sigma(s)$ satisfies the property U' then s satisfies the property U. This relation is associated with the reflection of properties, and we write it as $U \mathrel{-\!\!-\!\!<}^{f} U'$. For example, taking f to be a projection from a product, we can sometimes use these elementary ideas to show that a property holding of a component of a parallel composition entails a corresponding property holds of the composition, or vice versa.

Let U and U' be properties of transition systems T and T' and let $f\colon T \to T'$. As defined, the consideration of whether or not a relation $U \mathrel{-\!\!-\!\!>}^{f} U'$ or $U \mathrel{-\!\!-\!\!<}^{f} U'$ holds is determined by the existence or non-existence of an inclusion (thought of as entailment) in $\mathscr{P}(T)$. Because of the existence of adjoints to $\mathscr{P}f$, the truth of the relations is determined in $\mathscr{P}(T')$. Regarded as functor on the partial-order categories of properties, $\mathscr{P}f$ has a right and left adjoint $\forall_f, \exists_f\colon \mathscr{P}(T) \to \mathscr{P}(T')$, respectively, given by

$$\forall_f(U) = \{s' \text{ is reachable in } T' \mid \forall s.\sigma(s) = s' \Rightarrow s \in U\}$$

$$\exists_f(U) = \{s' \text{ is reachable in } T' \mid \exists s.\sigma(s) = s' \ \&\ s \in U\},$$

for any $U \in \mathscr{P}(T)$. The adjoints take the form of generalised quantifiers (see Poigné, 1985), which explains the notation. The adjunctions mean

$$\begin{aligned} &\mathscr{P}f(U') \subseteq U \Leftrightarrow U' \subseteq \forall_f(U), \\ &U \subseteq \mathscr{P}f(U') \Leftrightarrow \exists_f(U) \subseteq U', \end{aligned}$$

for all $U \in \mathscr{P}(T)$, $U' \in \mathscr{P}(T')$. As we have seen the relations involved in these equivalences have an intuitive significance. The relations $\mathscr{P}f(U') \subseteq U$ and $U' \subseteq \forall_f(U)$ are both equivalent to $U -\!\!<^{f} U'$. The relations $U \subseteq \mathscr{P}f(U')$ and $\exists_f(U) \subseteq U'$ are both equivalent to $U -\!\!>^{f} U'$.

Making use of the syntax of assertions we can investigate the preservation and reflection of assertions across morphisms $f = (\sigma, \lambda): T \to T'$ in general. Let $\phi: T$ and $\psi: T'$. We use $\phi -\!\!<^{f} \psi$ to mean $[\![\phi]\!] -\!\!<^{f} [\![\psi]\!]$ (reflection), and $\phi -\!\!>^{f} \psi$ to mean $[\![\phi]\!] -\!\!>^{f} [\![\psi]\!]$ (preservation). We see

$$\begin{aligned}
&I \overset{f}{-\!\!>} I \\
&\phi_0 \overset{f}{-\!\!>} \psi_0 \;\&\; \phi_1 \overset{f}{-\!\!>} \psi_1 \Rightarrow \phi_0 \wedge \psi_0 \overset{f}{-\!\!>} \psi_0 \wedge \psi_1 \\
&\phi_0 \overset{f}{-\!\!>} \psi_0 \;\&\; \phi_1 \overset{f}{-\!\!>} \psi_1 \Rightarrow \phi_0 \vee \psi_0 \overset{f}{-\!\!>} \psi_0 \vee \psi_1 \\
&\phi \overset{f}{-\!\!>} \psi \Rightarrow \langle a \rangle \phi \overset{f}{-\!\!>} \langle \lambda(a) \rangle \psi \\
&\phi \overset{f}{-\!\!>} \psi \Rightarrow \phi \langle a \rangle \overset{f}{-\!\!>} \psi \langle \lambda(a) \rangle \\
&\phi \overset{f}{-\!\!>} \psi \Leftrightarrow \neg \phi \overset{f}{-\!\!<} \neg \psi \\
&\phi_0 \overset{f}{-\!\!<} \psi_0 \;\&\; \phi_1 \overset{f}{-\!\!<} \psi_1 \Rightarrow \phi_0 \wedge \psi_0 \overset{f}{-\!\!<} \psi_0 \wedge \psi_1 \\
&\phi_0 \overset{f}{-\!\!<} \psi_0 \;\&\; \phi_1 \overset{f}{-\!\!<} \psi_1 \Rightarrow \phi_0 \vee \psi_0 \overset{f}{-\!\!<} \psi_0 \vee \psi_1 \\
&\phi \overset{f}{-\!\!<} \psi \Rightarrow [a] \phi \overset{f}{-\!\!<} [\lambda(a)] \psi \\
&\phi \overset{f}{-\!\!<} \psi \Rightarrow \phi [a] \overset{f}{-\!\!<} \psi [\lambda(a)].
\end{aligned}$$

In an informal sense, it appears that assertions expressing liveness properties are those which are preserved along morphisms (with some renaming), and that safety assertions are reflected.

Given a morphism $f: T \to T'$ of labelled transition systems the operations $\exists_f$, $\forall_f$, $\mathscr{P}f$ allow us to pass back and forth between properties of T and T'. The compositional proof system of the next section will make use of properties built up using $\exists_f$ and $\mathscr{P}f$ for particular morphisms f associated with our constructions on labelled transition systems. As an illustration we show how we can transfer properties between products and coproducts of transition systems and their components. The category **Prop** of transition-system properties has products and coproducts which motivate some extensions to our syntax of modal assertions. Let $U_0 : T_0$ and $U_1 : T_1$ be properties of labelled safe transition systems T_0 and T_1. Their product in **Prop** is a property of $T_0 \times T_1$, which we write as $U_0 \times U_1 : T_0 \times T_1$, where

$$U_0 \times U_1 = (\mathscr{P}\Pi_0)(U_0) \cap (\mathscr{P}\Pi_1)(U_1),$$

so $s \in U_0 \times U_1$ iff $\rho_0(s) \in U_0$ and $\rho_1(s) \in U_1$; the projections are $U_0 \times U_1 \xrightarrow{\Pi_k} U_k$ based on projections $\Pi_k : T_0 \times T_1 \to T_k$, for $k = 0, 1$, from the product of labelled transition systems. Their coproduct is a property $U_0 + U_1 : T_0 + T_1$, where

$$U_0 + U_1 = \exists_{In_0}(U_0) \cup \exists_{In_1}(U_1)$$

satisfied by states s of $T_0 + T_1$ iff $\iota_0(s) \in U_0$ or $\iota_1(s) \in U_1$. Extending the syntax for properties to support such products and coproducts of properties is the key to our compositional method of reasoning about parallel compositions and sums of transition systems.

6. A Compositional Proof System

We introduce a language of finite labelled safe transition systems. In presenting the language we assume finite sets of labels, the precise syntax of which we leave open, though it should support a notation for coproducts (disjoint unions) and products in the category of sets with partial functions. With one exception, the constructions in the language have already arisen as categorical constructions, and, for example, there are terms $t_0 + t_1$ and $t_0 \times t_1$ standing for the sum and product of labelled transition systems. As we have seen, such constructions have events which are labelled by elements of the sets $L_0 + L_1$ and $L_0 \times_* L_1$, where L_0 and L_1 label the events of the component transition systems. There is a term *nil* denoting a transition system with a single initial state (it is the initial object in our category of transition systems). Restrictions are denoted by terms of the form $t \lceil A$, where A is a subset of the labels L of the transition system T denoted by t; then $t \lceil A$ denotes the transition system $i^*(T)$, where $i: A \to L$. If t denotes a transition system T over labels L and $\Xi: L \to_* L'$ is total, then $t(\Xi)$ denotes the transition system $\Xi_!(T)$. Prefixing denoted by terms αt describes a transition system which which at first can perform only an α event and then behave as the transition system described by t. The remaining construct, taking the shape $t/\alpha, \phi$, introduces loops, in a way to be explained shortly. It is clear that each term is associated with a set of labels inherited from the labelled transition system it denotes. This we call the *sort* of the term.

A labelled-transition-system term t has a form given by the grammar

$$t ::= \mathrm{nil} \mid \alpha t \mid t_0 + t_1 \mid t_0 \times t_1 \mid t \lceil \Lambda \mid t(\Xi) \mid (t/\alpha, \phi),$$

where α is a label, Λ is a subset of labels of the sort of t, and Ξ in $t(\Xi)$ stands for a total function from sort(t) to a set of labels. We shall discuss the form of assertions to describe properties soon. For the moment it suffices to say that ϕ in t/α, ϕ denotes a property of the transition system denoted by t. Each transition system term is associated with a sort, the labelling set of its transition system:

$$\mathrm{sort}(nil) = \varnothing, \qquad \mathrm{sort}(\alpha t) = \{\alpha\} \cup \mathrm{sort}(t),$$
$$\mathrm{sort}(t_0 + t_1) = \mathrm{sort}(t_0) + \mathrm{sort}(t_1),$$
$$\mathrm{sort}(t_0 \times t_1) = \mathrm{sort}(t_0) \times_* \mathrm{sort}(t_1),$$
$$\mathrm{sort}(t \lceil \Lambda) = \mathrm{sort}(t) \backslash \Lambda, \qquad \mathrm{sort}(t(\Xi)) = \Xi\, \mathrm{sort}(t),$$
$$\mathrm{sort}(t/\alpha, J) = \mathrm{sort}(t) \cup \{\alpha\}.$$

It remains to explain the looping construction, represented by a term such as $t/\alpha, \phi$, which from the transition system denoted by t uses a label α and a property denoted by ϕ of t to adjoin loops. The term denotes a transition system which is that of t but with additional transitions labelled α which loop back from any state which satisfies ϕ. Its meaning is explained by the following construction on transition systems.

6.1. Definition. Let $T = (S, i, L, Tran)$ be a labelled transition system. Let U: T be a property of T. Let α be a label. The labelled transition system T/α, U is defined to be that transition system $(S, i, L, Tran')$, where

$$Tran' = Tran \cup \{(u, \alpha, i) \mid u \in U\}.$$

In other words, the transition system T/α, U is obtained from a transition system T by extending T by a new α transition from states in U to the initial state.

It is now straightforward to define $[\![t]\!]$, the labelled safe transition system denoted by a transition-system term t, taking, for instance, $[\![t_0]\!] \times [\![t_1]\!]$ as the denotation of $t_0 \times t_1$. The denotation of t/α, ϕ is that transition system obtained by introducing a loop from the reachable states specified by ϕ. Realise that for the moment we have not stated the precise form such syntax for ϕ can take. We omit the remainder of the otherwise obvious definition of the meaning of terms.

Each transition-system term possesses a rich language of assertions which determine properties. We use ϕ: t to indicate that ϕ is an assertion about the transition system denoted by t. And, of course, strictly speaking, the syntax of transition system terms and their assertions is given by mutual recursion, a term $t/\alpha, \phi$ being allowed if $\phi: t$. Each transition-system term has certain *basic* assertions determined by the way it is built up. More complicated assertions are formed by using propositional connectives, two modal operators and recursively defined assertions, using a least fixed point operator, in the way we have seen earlier. Typically, a term t is associated with assertions

$$\phi ::= \mathit{basic}\ |I|\ \mathrm{tt}\ |\mathrm{ff}|\ \phi_0 \wedge \phi_1\ |\phi_0 \wedge \phi_1|\ \neg\phi\ |\langle a\rangle\,\phi|\ \phi\langle a\rangle\,|X|\ \mu X.\phi,$$

where *basic* is the basic assertions of t, a is a label in sort(t) or $*$, and X is an assertion variable from Var. As before, we impose a condition on occurrences within the scope of μX in assertions $\mu X.\phi$ and shall use abreviations for implication and logical equivalence and the $[a]$-modalities.

As before, an assertion $\phi: t$ is satisfied by a subset of reachable states, its extension, and we could, for a more complete development, define the denotation $[\![\phi: t]\!]\vartheta$ of assertions ϕ of a transition-system term t, with respect to an environment ϑ, assigning properties to assertion variables. When ϕ is a closed assertion the denotation $[\![\phi: t]\!]\vartheta$ is independent of any environment ϑ, and we shall write its denotation as just $[\![\phi: t]\!]$.

It remains to define the basic assertions of each transition-system term. The idea is to import into a constructed transition system properties of its immediate subcomponents via the morphisms associated with the transition-system construction. As syntax for the basic assertions we adapt the transition-system constructors themselves, an idea which has been useful in several places (e.g., Hoare and Olderog, 1983; Winskel, 1985a; Brookes, 1985; Abramsky, 1987; Martin-Löf, 1983; Graf and Sifakis, 1985). To avoid complications with variables and environments we shall assume basic assertions are closed.

There are none for *nil*—its only atomic assertions are I, tt, and ff. Recalling 3.6 for the definition of prefixing, $\alpha\phi: \alpha t$ if $\phi: t$ and ϕ is closed, with

$$[\![\alpha\phi: \alpha t]\!] = \{\{s\} \mid s \in [\![\phi : t]\!]\},$$

which consists of those states of $[\![t]\!]$ which satisfy ϕ, but now regarded as states of $[\![\alpha t]\!]$.

$\phi_0 + \phi_1: t_0 + t_1$ if $\phi_0: t_0$ and $\phi_1: t_1$ and ϕ_0 and ϕ_1 are closed, with

$$[\![\phi_0 + \phi_1: t_0 + t_1]\!] = \exists_{In_0}[\![\phi_0: t_0]\!] \cup \exists_{\mathrm{In}_1}[\![\phi_1: t_0]\!],$$

which we recall from the last section is a coproduct in **Prop**.

$\phi_0 \times \phi_1: t_0 \times t_1$ if $\phi_0: t_0$ and $\phi_1: t_1$ and ϕ_0 and ϕ_1 are closed, with

$$[\![\phi_0 \times \phi_1: t_0 \times t_1]\!] = (\mathscr{P}\Pi_0)[\![\phi_0: t_0]\!] \cap (\mathscr{P}\Pi_1)[\![\phi_1: t_1]\!],$$

which we recall is a product in **Prop**.

$\phi\lceil\Lambda : t\lceil\Lambda$ if $\phi : t$ and ϕ is closed, with

$$[\![\phi\lceil\Lambda : t\lceil\Lambda]\!] = (\mathscr{P}i^*)[\![\phi : t]\!] = \{s \mid s \text{ is reachable in } [\![t\lceil\Lambda]\!] \;\&\; s \in [\![\phi : t]\!]\},$$

where $i^* : [\![t\lceil\Lambda]\!] \to [\![t]\!]$ is the cartesian lifting of the inclusion $i : \Lambda \to L$, which are all those reachable states of the restricted transition system which satisfy ϕ in the original transition system.

$\phi(\Xi) : t(\Xi)$ if $\phi : t$ and ϕ is closed, with

$$[\![\phi(\Xi) : t(\Xi)]\!] = (\exists_{\Xi_1})[\![\phi : t]\!] = \{s \mid s \in [\![\phi : t]\!]\}.$$

$(\phi/\alpha, J) : (t/\alpha, J)$ if $\phi : t$ and ϕ is closed, with

$$[\![\phi/\alpha, J : t/\alpha, J]\!] = \exists_j [\![\phi : t]\!] = \{s \mid s \in [\![\phi : t]\!]\},$$

where $j : [\![t]\!] \to [\![t/\alpha, J]\!]$ is the morphism expressing the fact that $[\![t]\!]$ is a subsystem of $[\![t/\alpha, J]\!]$.

We write $t \models \phi$, for a transition-system term t and closed assertion $\phi : t$, iff $[\![\phi]\!] = \{s \mid s$ is reachable in $[\![t]\!]\}$. More generally, we write $t, \Gamma \models \phi$, for $\Gamma = \{\theta_0 : t, ..., \theta_{n-1} : t\}$ a possibly empty set of closed assertions, iff $[\![\theta_0 \wedge \cdots \wedge \theta_{n-1} : t]\!] \subseteq [\![\phi : t]\!]$.

We can now complete the description of the syntax for the looping construct $t/\alpha, J$. Up till now we have not said how the property J is to be built up. In order to have a complete proof system we arrange that J denote a single reachable state. However, by building up J solely from constructors like product on assertions we obtain a notation for reachable states. For instance, if J_0 and J_1 are assertions satisfied by single reachable states of transition systems denoted by t_0 and t_1 then $J_0 \times J_1$ is satisfied by a single reachable state of their product and so can be taken to describe that reachable state. We give the rules to generate the notation for reachable states. It is intended that *singleton*$\{J : t\}$ holds iff an assertion J denotes a set consisting of exactly one reachable state of $[\![t]\!]$.

$$\mathit{singleton}\ \{I : t\} \qquad \frac{\mathit{singleton}\ \{J : t\}}{\mathit{singleton}\ \{\alpha J : \alpha t\}}$$

$$\frac{\mathit{singleton}\ \{J_0 : t_0\}}{\mathit{singleton}\ \{J_0 + \mathrm{ff} : t_0 + t_1\}} \qquad \frac{\mathit{singleton}\ \{J_1 : t_1\}}{\mathit{singleton}\ \{\mathrm{ff} + J_1 : t_0 + t_1\}}$$

$$\frac{\mathit{singleton}\ \{J_0 : t_0\},\ \mathit{singleton}\ \{J_1 : t_1\}}{\mathit{singleton}\ \{J_0 \times J_1 : t_0 \times t_1\}}$$

$$\frac{\mathit{singleton}\ \{J : t\}\ \ t \models J \to R_\Lambda}{\mathit{singleton}\ \{J\lceil\Lambda : t\lceil\Lambda\}}, \quad \text{where } R_\Lambda \equiv \mu X.\Big(I \vee \bigvee_{\alpha \in \Lambda} X\langle\alpha\rangle\Big)$$

$$\frac{\mathit{singleton}\ \{J : t\}}{\mathit{singleton}\ \{J(\Xi) : t(\Xi)\}} \qquad \frac{\mathit{singleton}\ \{J : t\}}{\mathit{singleton}\{J/\alpha, K : t/\alpha, K\}}.$$

As we have mentioned, strictly speaking the syntax of transition-system terms and their assertions is given by mutual recursion, a term $t/\alpha, J$ being allowed if *singleton* $\{J\colon t\}$. In turn the rules for *singleton* $\{J\colon t\}$ depend on the truth of assertions $t \models J \rightarrow R_A$ when a restriction is involved. We shall provide proof rules for this relation later and so correct the seeming dependence of syntax on a semantic notion. From earlier results characterising reachable states of constructed transition systems from those of their components we can justify the above rules by noting:

6.2. PROPOSITION. *Let t be a transition system term. If singleton$\{J\colon t\}$ then $[\![J\colon t]\!]$ denotes a singleton consisting of a single reachable state of $[\![t]\!]$. Conversely, if s is a reachable state of $[\![t]\!]$ then there is an assertion J for which singleton$\{J\colon t\}$ and $[\![J\colon t]\!] = \{s\}$.*

Now we present the proof rules associated with each operation on transition-system terms. These are adjoined to the proof rules introduced in Section 5 and displayed in Appendix II. We will have the result that, for a closed term t and assertion ϕ, we can prove $t \vdash \phi$ iff $t \models \phi$, which we recall means that $\phi\colon t$ is valid; i.e., *every* reachable state of t satisfies ϕ. More specifically, each term t will be associated with rules for deriving sequents $t, \Gamma \vdash \phi$ interpreted as $t, \Gamma \models \phi$. In Appendix II, there are rules for the modal μ-calculus, valid for any term t with the right sort (see Appendix II), and extra rules depending on the structure of t which express how each term operation interacts with the logical and modal operations. The latter rules will have the function of reducing the proof of a property of a compound term to proofs of properties of its immediate subcomponents. By *variable-free* assertions we mean those which do not contain any assertion variables, and so, in particular, can have no sub-assertions of the form $\mu X.\phi$; thus variable-free means closed and recursion free.

Product. The rules for the product play a fundamental role in proofs about parallel compositions. For terms t_0 and t_1, they use projection functions $\pi_k\colon \mathrm{sort}(t_0 \times t_1) \rightarrow_* \mathrm{sort}(t_k)$, for $k = 0, 1$.

$$t_0 \times t_1 \vdash I \leftrightarrow I \times I$$

$$t_0 \times t_1 \vdash \mathrm{tt} \leftrightarrow \mathrm{tt} \times \mathrm{tt}$$

$$t_0 \times t_1 \vdash \mathrm{ff} \leftrightarrow ([\mathrm{ff} \times \mathrm{tt}] \vee [\mathrm{tt} \times \mathrm{ff}])$$

$$t_0 \times t_1 \vdash ([\phi_0 \times \phi_1] \wedge [\phi_0' \times \phi_1']) \leftrightarrow ([\phi_0 \wedge \phi_0'] \times [\phi_1 \wedge \phi_1'])$$

$$t_0 \times t_1 \vdash ([\phi_0 \times \phi_1] \vee [\phi_0' \times \phi_1]) \leftrightarrow [(\phi_0 \vee \phi_0') \times \phi_1]$$

$$t_0 \times t_1 \vdash ([\phi_0 \times \phi_1] \vee [\phi_0 \times \phi_1']) \leftrightarrow [\phi_0 \times (\phi_1 \vee \phi_1')]$$

$$t_0 \times t_1 \vdash \neg [\phi_0 \times \phi_1] \leftrightarrow ([\neg \phi_0 \times \mathrm{tt}] \vee [\mathrm{tt} \times \neg \phi_1])$$
$$t_0 \times t_1 \vdash \langle a \rangle [\phi_0 \times \phi_1] \leftrightarrow [\langle \pi_0(a) \rangle \phi_0 \times \langle \pi_1(a) \rangle \phi_1]$$
$$t_0 \times t_1 \vdash [\phi_0 \times \phi_1] \langle a \rangle \leftrightarrow [\phi_0 \langle \pi_0(a) \rangle \times \phi_1 \langle \pi_1(a) \rangle]$$
$$\frac{t_0 \vdash \phi_0 \quad t_1 \vdash \phi_1}{t_0 \times t_1 \vdash \phi_0 \times \phi_1}.$$

The purpose of these rules is to reduce the proof that an assertion is valid of a product to the proof that certain assertions are valid in its components. First, all assertions concerning a product can be put into a normal form:

6.3. Lemma. *Let $\phi: t_0 \times t_1$ be an assertion which is variable-free. Then*

$$t_0 \times t_1 \vdash \phi \leftrightarrow \bigvee_{k \in K} \phi_{0k} \times \phi_{1k},$$

where K is a finite set, indexing variable-free assertions $\phi_{0k}: t_0$ and $\phi_{1k}: t_1$ for $k \in K$.

Proof. The lemma follows by structural induction on ϕ using basic distributivity properties of the logical connectives. For example, to deal with the hardest case of the induction, we show that if we assume the proposition holds for assertion ϕ then it holds for assertion $\neg \phi$.

Suppose $t_0 \times t_1 \vdash \phi \leftrightarrow \bigvee_{i \in I} \phi_{0i} \times \phi_{1i}$. Then

$$t_0 \times t_1 \vdash \neg \phi \leftrightarrow \bigwedge_{i \in I} \neg\ [\phi_{0i} \times \phi_{1i}],$$

and so

$$t_0 \times t_1 \vdash \neg \phi \leftrightarrow \bigwedge_{i \in I} ([\neg \phi_{0i} \times \mathrm{tt}] \vee [\mathrm{tt} \times \neg \phi_{1i}]).$$

But then $t_0 \times t_1 \vdash \neg \phi \leftrightarrow \bigvee W$, where W is the set of product assertions

$$\left\{ \left[\bigwedge_{i \in I} \phi^*_{0i} \times \bigwedge_{i \in I} \phi^*_{1i} \right] \middle| \begin{array}{l} (\phi^*_{0i} \equiv \neg \phi_{0i} \;\&\; \phi^*_{1i} \equiv \mathrm{tt}) \\ \text{or} \quad (\phi^*_{0i} \equiv \mathrm{tt} \;\&\; \phi^*_{1i} \equiv \neg \phi_{1i}), \text{ all } i \in I \end{array} \right\},$$

making ϕ provably equivalent to an assertion of the right form. ∎

Now it is shown that an assertion in normal form, valid of a product, has a factorisation into two assertions, valid of the respective components, whose product provably entails the original assertion:

6.4. Lemma (The factorisation lemma). *Let t_0 and t_1 be transition-system terms such that*

$$t_0 \times t_1 \models \bigvee_{k \in K} \phi_{0k} \times \phi_{1k},$$

for variable-free assertions ϕ_{0k}, ϕ_{1k}. Then there are variable-free assertions $\phi_0 : t_0$ and $\phi_1 : t_1$ for which

$$t_0 \models \phi_0 \qquad \& \qquad t_1 \models \phi_1 \qquad \& \qquad t_0 \times t_1, \phi_0 \times \phi_1 \vdash \bigvee_{k \in K} \phi_{0k} \times \phi_{1k}.$$

Proof. We argue that ϕ_0 and ϕ_1 exist in a nonconstructive way which shows that there exists a proof, so $t_0 \times t_1, \phi_0 \times \phi_1 \vdash \bigvee_{i \in I} \phi_{0i} \times \phi_{1i}$, without our giving it explicitly. We know that $t_0 \times t_1 \models \bigvee_{i \in I} \phi_{0i} \times \phi_{1i}$. Hence there is a function $i[\ ,\ ]$ so that for any reachable states $x : [\![t_0]\!]$ and $y : [\![t_1]\!]$ there is $i[x, y] \in I$ such that

$$x \in [\![\phi_{0_{i[x,y]}}]\!] \qquad \text{and} \qquad y \in [\![\phi_{1_{i[x,y]}}]\!].$$

(Note that this does not determine $i[\,,]$ uniquely.) For notational convenience write $x : t_0$ and $y : t_1$ to mean x is a reachable state of $[\![t_0]\!]$ and y is a reachable state of $[\![t_1]\!]$ for the duration of this proof. Now

$$x \models \bigwedge_{y : t_1} \phi_{0_{i[x,y]}} \qquad \& \qquad y \models \bigwedge_{x : t_0} \phi_{1_{i[x,y]}}$$

for any $x : t_0$ and $y : t_1$. We use, e.g., $\bigwedge_{y : t_1} \phi_{0_{i[x,y]}}$ to mean the finite conjunction

$$\bigwedge \{\phi_{0_{i[x,y]}} \mid y : t_1\}.$$

Clearly

$$t_0, \bigwedge_{y : t_1} \phi_{0_{i[x,y]}} \vdash \phi_{0_{i[x,y]}} \qquad \text{for} \quad x : t_0 \qquad \text{and} \qquad y : t_1,$$

and

$$t_1, \bigwedge_{x : t_0} \phi_{1_{i[x,y]}} \vdash \phi_{1_{i[x,y]}} \qquad \text{for} \quad x : t_0 \qquad \text{and} \qquad y : t_1,$$

as, e.g., the conjunction $\bigwedge_{y : t_1} \phi_{0_{i[x,y]}}$ contains $\phi_{0_{i[x,y]}}$ as a conjunct. Thus

$$t_0 \times t_1, \bigwedge_{y : t_1} \phi_{0_{i[x,y]}} \times \bigwedge_{x : t_0} \phi_{1_{i[x,y]}} \vdash \phi_{0_{i[x,y]}} \times \phi_{1_{i[x,y]}}$$

for any $x : t_0$ and $y : t_1$, Write

$$\phi_0 \equiv \bigvee_{x : t_0} \bigwedge_{y : t_1} \phi_{0_{i[x,y]}} \qquad \text{and} \qquad \phi_1 \equiv \bigvee_{y : t_1} \bigwedge_{x : t_0} \phi_{1_{i[x,y]}}.$$

Using the rules expressing the fact that × distributes over ∨ we obtain

$$t_0 \times t_1, \phi_0 \times \phi_1 \vdash \bigvee\!\!\!\bigvee_{x:t_0,\, y:t_1} \left[\bigwedge\!\!\!\bigwedge_{y:t_1} \phi_{0_{i[x,y]}} \times \bigwedge\!\!\!\bigwedge_{x:t_0} \phi_{1_{i[x,y]}} \right].$$

Also, obviously,

$$t_0 \times t_1 \bigvee\!\!\!\bigvee_{x:t_0,y:t_1} \left[\bigwedge\!\!\!\bigwedge_{y:t_1} \phi_{0_{i[x,y]}} \times \bigwedge\!\!\!\bigwedge_{x:t_0} \phi_{1_{i[x,y]}} \right] \vdash \bigvee\!\!\!\bigvee_{i \in I} \phi_{0_i} \times \phi_{1_i}.$$

Therefore $t_0 \times t_1, \phi_0 \times \phi_1 \vdash \bigvee\!\!\!\bigvee_{i \in I} \phi_{0i} \times \phi_{1i}$. Clearly $t_0 \models \phi_0$ and $t_1 \models \phi_1$. ▌

The importance of the above two lemmas is that they reduce the problem of proving a variable-free assertion ϕ holds of a transition system term $t_0 \times t_1$ to showing assertions $\phi_0 \colon t_0$ and $\phi_1 \colon t_1$ hold of the components. Once $t_0 \vdash \phi_0$ and $t_1 \vdash \phi_1$ are established, from the rule

$$\frac{t_0 \vdash \phi_0 \; t_1 \vdash \phi_1}{t_0 \times t_1 \vdash \phi_0 \times \phi_1}$$

it can be concluded that $t_0 \times t_1 \vdash \phi_0 \times \phi_1$ and hence $t_0 \times t_1 \vdash \phi$.

Restriction. The following rules reduce the proof of the validity of an assertion $\psi \colon t \lceil A$ to the validity of an assertion $\phi \colon t$.

$$t \lceil A \vdash I \leftrightarrow I \lceil A$$

$$t \lceil A \vdash \mathrm{tt} \leftrightarrow \mathrm{tt} \lceil A, \qquad t \lceil A \vdash \mathrm{ff} \leftrightarrow \mathrm{ff} \lceil A,$$

$$t \lceil A \vdash \neg(\phi \lceil A) \leftrightarrow (\neg \phi \lceil A)$$

$$t \lceil A \vdash (\phi_0 \lceil A) \wedge (\phi_1 \lceil A) \leftrightarrow (\phi_0 \wedge \phi_1) \lceil A$$

$$t \lceil A \vdash (\phi_0 \lceil A) \vee (\phi_1 \lceil A) \leftrightarrow (\phi_0 \vee \phi_1) \lceil A$$

$$t \lceil A \vdash \langle a \rangle (\phi \lceil A) \leftrightarrow (\langle a \rangle \phi) \lceil A, \qquad \text{where} \quad a \in A \cup \{*\},$$

$$t \lceil A \vdash (\phi \lceil A) \langle a \rangle \leftrightarrow (\phi \wedge R_A) \langle a \rangle \lceil A, \qquad \text{where} \quad a \in A \cup \{*\},$$

$$\frac{t \vdash R_A \to \phi}{t \lceil A \vdash \phi \lceil A}, \qquad \text{where} \quad R_A \equiv \mu.\Delta I \vee \bigvee\!\!\!\bigvee_{a \in A} X \langle \alpha \rangle \Big).$$

6.5. Lemma. *For all variable-free $\psi \colon t \lceil A$ there is a closed assertion $\phi \colon t$ such that $t \lceil A \vdash \psi \leftrightarrow \phi \lceil A$.*

Proof. By structural induction on ψ using the rules for restriction. ▌

In this way establishing $t \lceil A \vdash \psi$ is reduced to that of establishing $t \lceil A \vdash \phi \lceil A$. But $t \lceil A \models \phi \lceil A$ iff $t \models (R_A \to \phi)$, where

$R_A \equiv \mu X.(I \vee \bigvee_{\alpha \in A} X\langle\alpha\rangle)$. Once we have proved the appropriate validity for t, applying the rule

$$\frac{t \vdash R_A \rightarrow \phi}{t\lceil A \vdash \phi\lceil A}$$

gives the desired validity for $t\lceil A$.

Relabelling. The following rules reduce the proof that $\psi : t(\Xi)$ is valid to the proof of the validity of an assertion $\phi : t$.

$$t(\Xi) \vdash I \leftrightarrow I(\Xi)$$

$$t(\Xi) \vdash \mathrm{tt} \leftrightarrow \mathrm{tt}(\Xi), \qquad t(\Xi) \vdash \mathrm{ff} \leftrightarrow \mathrm{ff}(\Xi),$$

$$t(\Xi) \vdash \neg(\phi(\Xi)) \leftrightarrow (\neg\phi(\Xi))$$

$$t(\Xi) \vdash \phi_0(\Xi) \wedge \phi_1(\Xi) \leftrightarrow (\phi_0 \wedge \phi_1)(\Xi)$$

$$t(\Xi) \vdash \phi_0(\Xi) \vee \phi_1(\Xi) \leftrightarrow (\phi_0 \vee \phi_1)(\Xi)$$

$$t(\Xi) \vdash \langle a\rangle(\phi(\Xi)) \leftrightarrow \left(\bigvee_{b \in \Xi^{-1}a} \langle b\rangle\phi\right)(\Xi),$$

$$t(\Xi) \vdash (\phi(\Xi))\langle a\rangle \leftrightarrow \left(\bigvee_{b \in \Xi^{-1}a} \phi\langle b\rangle\right)(\Xi),$$

$$\frac{t \vdash \phi}{t(\Xi) \vdash \phi(\Xi)},$$

6.6. LEMMA. *For all variable-free $\psi : t(\Xi)$ there is a variable-free assertion $\phi : t$ such that $t(\Xi) \vdash \psi \leftrightarrow \phi(\Xi)$.*

Proof. By structural induction on ψ using the rules for relabelling. ▮

In this way, establishing the validity of $\psi : t(\Xi)$ is reduced to establishing that of $\phi(\Xi) : t(\Xi)$, for an assertion $\phi : t$. But this amounts to establishing the validity of $\phi : t$, as is supprted by the rule

$$\frac{t \vdash \phi}{t(\Xi) \vdash \phi(\Xi)}.$$

To summarise, the rules for the operations associated with parallel composition have a straightforward nature. The rules for restriction express directly how routinely to convert an assertion $\psi : t\lceil A$ to one $\phi\lceil A : t\lceil A$ which has the same extension. In this way the proof of validity of $\psi : t\lceil A$ is reduced to that of $R_A \rightarrow \phi : t$. The rules for relabelling play a very similar role. Those for product have to cater for the complication that a property of a product cannot in general be expressed in the form of a single product of properties $U_0 \times U_1$. Their form is dictated by the requirement of displaying a property of a product as a disjunction of products of properties.

Unfortunately, the rules for sum, looping, and prefixing especially are more ad hoc. This is largely due to the special account that must be taken of the initial state in reasoning about the these constructions. Again, however, the motivating idea is to express the extension of an assertion of these constructions in terms of assertions about the immediate components.

Sum. For the sum construction we provide rules so that an assertion $\psi: t_0 + t_1$ can be proved equivalent to one of the form $\phi_0 + \phi_1 : t_0 + t_1$. Recall that the states of $t_0 + t_1$ are copies of states of t_0 and t_1, disjoint but for being identified at the initial states. The extension of an assertion $\phi_0 + \phi_1 : t_0 + t_1$ is the union of the copies of the extensions of $\phi_0 : t_0$ and $\phi_1 : t_1$. As such they may or may not overlap at the identified initial states. Because of this it is not necessarily the case that $\neg(\phi_0 + \phi_1) : t_0 + t_1$ has the same extension as $(\neg\phi_0) + (\neg\phi_1) : t_0 + t_1$—consider, for example, taking $\phi_0 \equiv \mathrm{tt}$ and $\phi_1 \equiv \mathrm{ff}$. The rule for negation depends for its application on an assertion $\phi_0 + \phi_1 : t_0 + t_1$ being *balanced*. This is so when $t_0 \vdash I \to \phi_0$ iff $t_1 \vdash I \to \phi_1$, corresponding to times at which the component assertions are either both true or both false at the initial states. The rule for conjunction is similar. The rules for negation and conjunction have the form

$$\frac{\phi_0 + \phi_1 \text{ balanced}}{t_0 + t_1 \vdash \neg(\phi_0 + \phi_1) \leftrightarrow (\neg\phi_0) + (\neg\phi_1)}$$

$$\frac{\phi_0 + \phi_1,\ \phi_0' + \phi_1' \text{ balanced}}{t_0 + t_1 \vdash ((\phi_0 + \phi_1) \wedge (\phi_0' + \phi_1')) \leftrightarrow ((\phi_0 \wedge \phi_0') + (\phi_1 \wedge \phi_1'))}.$$

The fact that sums identify initial states of their components complicates the rules for modalities too. We use the notation $j_k : \mathrm{sort}(t_k) \to_* \mathrm{sort}(t_0 + t_1)$, $k = 0, 1$, for injections on label sets. Consider putting an assertion $\langle j_0(a_0)\rangle(\phi_0 + \phi_1)$, where $a_0 \in \mathrm{sort}(t_0)$, into the form of a sum of assertions. Whether $\langle j_0(a_0)\rangle(\phi_0 + \phi_1)$ is equivalent to $(\langle a_0\rangle(\phi_0 \vee I) + \mathrm{ff})$ or $(\langle a_0\rangle\, \phi_0 + \mathrm{ff})$ depends on whether or not ϕ_1 holds at the initial state of t_1. This explains why, for the forwards modality, we have the two rules

$$\frac{t_1 \vdash I \to \phi_1}{t_0 + t_1 \vdash \langle j_0(a_0)\rangle(\phi_0 + \phi_1) \leftrightarrow (\langle a_0\rangle(\phi_0 \vee I) + \mathrm{ff})}$$

$$\frac{t_1 \vdash I \to \neg\phi_1}{t_0 + t_1 \vdash \langle j_0(a_0)\rangle(\phi_0 + \phi_1) \leftrightarrow (\langle a_0\rangle\, \phi_0 + \mathrm{ff})}.$$

There are, of course, similar rules for the forwards modalities, involving labels like $j_1(a_1)$, as well as rules for the backwards modality, making a total of eight rules in all to handle the modalities. A further rule introduces valid sums of assertions. These and the other rules for reasoning about assertions for sums are given in Appendix II.

The essential facts provided by the rules are summarised in the following lemma. It shows how, on the assumption that the proof system is complete for subterms, assertions for a sum can be provably replaced by balanced assertions, and that, in general, such an assertion can be proved equivalent to a sum of assertions.

6.7. Lemma. *Let t_0, t_1 be transition system terms. Assume*

$$(t_0 \vdash \phi_0 \text{ iff } t_0 \models \phi_0) \quad \text{and} \quad (t_1 \vdash \phi_1 \text{ iff } t_1 \models \phi_1)$$

for all $\phi_0 : t_0$ and $\phi_1 : t_1$.

(i) *Let $\phi_0 : t_0$ and $\phi_1 : t_1$ be variable-free. Then there is a balanced variable-free assertion $\phi'_0 + \phi'_1$ such that*

$$t_0 + t_1 \vdash (\phi_0 + \phi_1) \leftrightarrow (\phi'_0 + \phi'_1).$$

(ii) *Let $\phi : t_0 + t_1$ be a variable-free assertion. Then there are variable-free $\phi_0 : t_0$ and $\phi_1 : t_1$ such that*

$$t_0 + t_1 \vdash \phi \leftrightarrow (\phi_0 + \phi_1).$$

(iii) *For variable-free $\psi : t_0 + t_1$,*

$$t_0 + t_1 \vdash \psi \quad \text{iff} \quad t_0 + t_1 \models \psi.$$

Proof. (i) With the first few rules for sum in Appendix II it can be proved that

$$t_0 + t_1 \vdash ((\phi_0 \vee I) + \phi_1) \leftrightarrow (\phi_0 + (\phi_1 \vee I)).$$

Using this, any unbalanced assertion can be proved equivalent to a balanced one.

(ii) By structural induction any variable-free assertion $\psi : t_0 + t_1$ can be proved equivalent to one of the form $\phi_0 + \phi_1$. In the case of modalities this reduction depends on what the initial state satisfies in the subterms t_0 and t_1. In the case of conjunctions and negations it depends on immediate subassertions of ψ having been previously proved equivalent to balanced assertions. ▌

Prefixing. In the case of prefixing we expect only $\psi : \alpha t$ to be equivalent to one of the form $\alpha\phi : \alpha t$ at non-initial states. The rules for prefixing, presented in Appendix II, are designed to give a method for replacing an assertion $\psi : \alpha t$ by an equivalent one $\alpha\phi$ assuming $\neg I$. They are admittedly somewhat ad hoc. As the lemma sketches, they are sufficient to prove

$$\alpha t, \neg I \vdash \psi \leftrightarrow \alpha\phi$$

and the appropriate one of

$$\alpha t, I \vdash \psi \qquad \text{or} \qquad \alpha t, I \vdash \neg\psi$$

on the assumption that the proof system is complete for t.

6.8. LEMMA. *Let t be a transition system term. Assume*

$$t \vdash \phi \qquad \textit{iff} \quad t \models \phi$$

for all variable-free $\phi : t$.

(i) *Let $\psi : \alpha t$ be a variable-free assertion. Then there is a variable-free assertion $\phi : t$ such that*

$$\alpha t, \neg I \vdash (\psi \leftrightarrow \alpha\phi).$$

(ii) *Let $\psi : \alpha t$ be a variable-free assertion. Then*

$$\alpha t, I \vdash \psi \qquad \textit{or} \qquad \alpha t, I \vdash \neg\psi.$$

(iii) *For variable-free $\psi : \alpha t$,*

$$\alpha t \vdash \psi \qquad \textit{iff} \quad \alpha t \models \psi.$$

Proof. It is a routine matter to show that the rules for prefixing given in Appendix II are sound. The conjunction of (i) and (ii), taken as an induction hypothesis, is proved by structural induction on ψ. (The proof of (i) and (ii) has an elementary, if involved and technical, character similar to that of (iii) below, and is omitted.) Soundness of the rules gives the "only if" direction of the proof of part (iii). The "if" direction follows from (i) and (ii) in the following way: Assume $\alpha t \models \psi$. Then by (i) and (ii) we obtain

$$\alpha t, \neg I \vdash (\psi \leftrightarrow \alpha\phi) \qquad \text{and} \qquad \alpha t, I \vdash \psi.$$

Soundness of the rules gives $\alpha t, \neg I \models (\psi \leftrightarrow \alpha\phi)$, from which it follows that $t \models \phi$ so $t \vdash \phi$ by the assumptions of the lemma. Thus $t \vdash \phi$ so $t \vdash \text{tt} \rightarrow \phi$. The rule gives

$$\alpha t \vdash \alpha\, \text{tt} \rightarrow \alpha\phi.$$

However, by the first two rules we can derive $\alpha t \vdash \neg I \leftrightarrow \alpha\mathrm{tt}$ and hence

$$\alpha t, \neg I \vdash \psi.$$

With $\alpha t, I \vdash \psi$, established earlier, we obtain $\alpha t \vdash \psi$ as required. ▌

Looping. The rules for the looping construction, given in Appendix II, are sufficient to reduce an assertion $\psi: t/\alpha, J$ to an equivalent assertion $\phi/\alpha, J: t/\alpha, J$. There is then a rule to introduce such assertions when valid. The rules are similar to those for sum in that, like sum, the reductions for a modal assertion $\langle\alpha\rangle(\phi/\alpha, J): t/\alpha, J$, or $(\phi/\alpha, J)\langle\alpha\rangle: t/\alpha, J$, depend on whether or not the initial state, or state specified by J, satisfies ϕ. For example, consider an assertion $(\phi/\alpha, J)\langle\alpha\rangle$. This is satisfied by all states which can be reached via an α transition from one satisfying ϕ in the original transition system t. There are thus two ways in which the initial state can satisfy $(\phi/\alpha, J)\langle\alpha\rangle$; one is through the initial state of t satisfying $\phi\langle\alpha\rangle$ and the other through the state specified by J satisfying ϕ—remember that the looping construction introduces an α transition from this state back to the initial state. This accounts for the two rules:

$$\frac{t \vdash J \rightarrow \neg\phi}{t/\alpha, J \vdash (\phi/\alpha, J)\langle\alpha\rangle \leftrightarrow (\phi\langle\alpha\rangle)/\alpha, J}$$

$$\frac{t \vdash J \rightarrow \phi}{t/\alpha, J \vdash (\phi/\alpha, J)\langle\alpha\rangle \leftrightarrow (\phi\langle\alpha\rangle \leftrightarrow (\phi\langle\alpha\rangle \vee I)/\alpha, J}.$$

Note that if we did not insist that J specify a single state these two rules would not cover the possibility of some, but not all, states specified by J satisfying ϕ. In this situation, as well, we would like to be able to deduce that $(\phi/\alpha, J)\langle\alpha\rangle$ was equivalent to $(\phi\langle\alpha\rangle \vee I)/\alpha, J$. However, to cope with this case it seems we would need to make use of the extra judgement that $J \wedge \phi$ is *satisfiable* by a reachable state of t (or, equivalently, $\neg(J \wedge \phi)$ is *not* valid). This judgement is not available in the present proof system.

6.9. LEMMA. *Let t be a transition system term. Assume*

$$t \vdash \phi \quad \textit{iff} \quad t \models \phi$$

for all variable-free $\phi: t$.

(i) *Let $\psi: t/\alpha, J$ be a variable-free assertion. Then there is a variable-free assertion $\phi: t$ such that*

$$t/\alpha, J \vdash \{\psi \leftrightarrow \phi/\alpha, J).$$

(ii) *For variable-free $\psi: t/\alpha, J$,*

$$t/\alpha, J \vdash \psi \quad \textit{iff} \quad t/\alpha, J \models \psi.$$

Proof. The proof of (i) is by structural induction on ψ using the rules for looping of Appendix II. Soundness and application of the final rule gives (ii). ▮

The previous lemmas, 6.3 to 6.9, reduce the validity of a variable-free assertion to the validity of assertions for subterms, though the latter need not be variable-free because of the rules for restriction. However, in general, we can eliminate recursion from closed assertions:

6.10. Lemma. *Let t be a transition-system term. For any $\phi : t$, a closed assertion, there is a variable-free assertion $\phi' : t$ such that $t \vdash (\phi \leftrightarrow \phi')$.*

Proof. Because the transition systems denoted by terms are finite, with only finite sets of reachable states, any recursively defined assertion is equivalent to some finite unfolding. Hence by structural induction on closed assertions for a term we can show that they are provably equivalent to variable-free assertions. ▮

Ultimately the validity of any assertion can be reduced to that of assertions for *nil.* We can show nil $\vdash \phi$ iff nil $\Vdash \phi$, so by structural induction on terms, combining the lemmas, we obtain the main result of this section. The proof rules are sound and complete for all closed assertions including those defined recursively.

6.11. Theorem (soundness and completeness). *The rules of Appendix* II *provide a sound and complete proof system, establishing $t \vdash \phi$ iff $t \models \phi$ for any closed assertion $\phi : t$.*

7. Conclusion

An extension of the work of Winskel (1984, 1987a) to include a category-theoretic treatment of event labels has been presented. The understanding of everything here as constructions in category theory is clearly incomplete: the prefixing and looping constructions are not characterised as universal constructions or explained categorically, and the modalities of the logic and some of the rules were produced in an a hoc manner. The use of other, categorical constructions has not been fully explored; for instance, as was remarked, another looping construction can be obtained from $\lambda_!$ once there is a way to put a transition between the two states to be identified.

At least some results have been laid down, and the categorical view of models for parallelism has been shown to have promise. Several other models can be related in the same way. Essentially the same ideas go through for labelled Petri nets. Results analogous to the relation between labelled transition systems and synchronisation trees hold between labelled safe Petri nets and labelled event structures. There is also a reflection between a category of sets of Hoare traces and synchronisation trees. However, the analogous relationship between labelled event structures and Pratt's (1986) pomsets is more subtle.

Whether or not the operation of hiding (that of making certain transitions/events internal) can be found a category-theoretic expression remains to be seen. The naive idea of taking partial labelling functions on transitions/events will not do, at least, not without a revision of the way parallel compositions are treated (see the Acknowledgments). It is likely that a more refined analysis of the concept of hiding will be needed first and that this will lead to more structure in the category of labelling sets.

On the more practical side, it is to be hoped that compositional proof systems like that above will be useful for verifying properties of parallel processes, perhaps in model checking (the automation of correctness proofs for finite state processes). By breaking the verification problem down into smaller subproblems it may extend the range of model checking, or at least provide a way to proceed in model checking which follows the structure of the design. This could involve the user's supplying assertions which are believed to hold of the components of a parallel composition. For all the constructions but product the proof rules supply an automatic procedure for reducing the problem of whether or not an assertion is valid of a compound term to whether or not assertions are valid in its immediate components. Indeed, for product-free terms, the rules as they stand can be used to decide validity, so that the rules for looping can be amended to cope with the construction $t/\alpha, \phi$, where ϕ does not denote a singleton. Despite the unattractiveness of some of the rules they code quite neatly in Standard ML to give a reduction of the validity of nonrecursive assertions for all but products to that in the component processes. Because of the nonconstructive nature of its proof, the "factorisation lemma," 6.4, does not directly yield a method for decomposing the validity of an assertion about a product to validities of its components. However, there is a related method for deciding validities of products, at least for nonrecursive assertions. Its feasibility, along with the problem of how to incorporate recursively defined assertions into such compositional model checking, is a topic currently under study.

It remains to extend the proof system to recursively defined processes where transition-system terms include an operation $rec\ x \cdot t$ instead of a simple looping construct $t/\alpha, J$, along the lines of Stirling (1985) and Winskel (1985a). Transition-system semantics can be given to such recursive definitions. But obtaining a compositional proof system for the assertion language is not a simple matter; current work indicates it may best be done by using an intuitionistic logic on assertions. A stumbling block to a full semantic treatment of this is that transition systems, and other catgories of models, like synchronisation trees, do not have the structure to support the notions of convergence and divergence as were used by, e.g., Stirling (1985) and Winskel (1985a).

APPENDIX I: Partial Functions

We shall work with a particular representation of the category of sets with partial functions. Assume that X and Y are sets not containing the distinguished symbol $*$. Write $f: X \rightarrow_* Y$ for a function $f: X \cup \{*\} \rightarrow Y \cup \{*\}$ such that $f(*) = *$. When $f(x) = *$, for $x \in X$, we say $f(x)$ is *undefined* and otherwise *defined*. We say $f: X \rightarrow_* Y$ is total when $f(x)$ is defined for all $x \in X$. Of course, such total morphisms $X \rightarrow_* Y$ correspond to the usual total functions $X \rightarrow Y$, with which they shall be identified. For the category $\mathbf{Set}_*$, we take as objects sets which do not contain $*$, and as morphisms functions $f: X \rightarrow_* Y$; the composition of two such functions is the usual composition of total functions (but on sets extended by $*$). Of course, $\mathbf{Set}_*$ is equivalent to the category of sets with partial functions, as usually presented.

We remark on two categorical constructions in $\mathbf{Set}_*$. A coproduct of X and Y in $\mathbf{Set}_*$ is the disjoint union $X + Y$ with the obvious injections. A product of X and Y in $\mathbf{Set}_*$ has the form

$$X \times_* Y = \{(x, *) \mid x \in X\} \cup \{(*, y) \mid y \in Y\} \cup \{(x, y) \mid x \in X, y \in Y\}$$

with projections of those partial functions to the left and right coordinates.

APPENDIX II: Proof Rules

The logic works on sequents of the form $t, \Gamma \vdash \phi$, where t is a term denoting a labelled transition system, Γ is a finite set of assertions, and $\phi: t$ is an assertion denoting properties of the transition system. Take $\phi \rightarrow \psi$ to abbreviate $\neg\phi \vee \psi$, and $\phi \leftrightarrow \psi$ to abbreviate $\phi \rightarrow \psi \wedge \psi \rightarrow \phi$. Let $\Gamma = \{\phi_0, ..., \phi_n\}$ be a finite set of assertions. We use $\bigwedge\!\!\!\bigwedge \Gamma$ to abbreviate their conjunction $\phi_0 \wedge \cdots \wedge \phi_n$ and $\bigvee\!\!\!\bigvee \Gamma$ to abbreviate their disjunction $\phi_0 \vee \cdots \vee \phi_n$. We identify $\bigwedge\!\!\!\bigwedge \varnothing$ with tt and $\bigvee\!\!\!\bigvee \varnothing$ with ff.

Rules for the Modal Logic

Structural Rules

$$(\text{refl}) \quad t, \Gamma \vdash \phi \quad \text{if} \quad \phi \in \Gamma, \qquad (\text{tran}) \quad \frac{t, \Gamma \vdash \phi \quad t, \Delta, \phi \vdash \psi}{t, \Gamma, \Delta \vdash \psi}.$$

Propositional Logic

$$(\wedge I) \qquad \frac{t, \Gamma \vdash \phi \quad t, \Gamma \vdash \psi}{t, \Gamma \vdash \phi \wedge \psi},$$

$$(\vee I) \qquad \frac{t, \Gamma \vdash \phi}{t, \Gamma \vdash \phi \wedge \psi} \quad \frac{t, \Gamma \vdash \psi}{t, \Gamma \vdash \phi \vee \psi},$$

$$(\rightarrow I) \qquad \frac{t, \Gamma, \phi \vdash \psi}{t, \Gamma \vdash \phi \rightarrow \psi},$$

$$(\neg I) \qquad \frac{t, \Gamma, \phi \vdash \mathrm{ff}}{t, \Gamma \vdash \neg \phi},$$

$$(\mathrm{tt}) \qquad t, \Gamma \vdash \mathrm{tt},$$

$$(\wedge E) \qquad \frac{t, \Gamma \vdash \phi \wedge \psi}{t, \Gamma \vdash \phi} \quad \frac{t, \Gamma \vdash \phi \wedge \psi}{t, \Gamma \vdash \psi},$$

$$(\vee E) \qquad \frac{t, \Gamma \vdash \phi \vee \psi \quad t, \Gamma, \phi \vdash \theta \quad t, \Gamma, \psi \vdash \theta}{t, \Gamma \vdash \theta},$$

$$(\rightarrow E) \qquad \frac{t, \Gamma \vdash \phi \rightarrow \psi \quad t, \Gamma \vdash \phi}{t, \Gamma \vdash \psi},$$

$$(\neg E) \qquad t, \Gamma, \phi, \neg \phi \vdash \mathrm{ff},$$

$$(\mathrm{ff}) \qquad \frac{t, \Gamma, \neg \phi \vdash \mathrm{ff}}{t, \Gamma \vdash \phi}.$$

Modal Rules

$$t \vdash \langle a \rangle\, \mathrm{ff} \leftrightarrow \mathrm{ff}, \qquad t \vdash (\langle a \rangle\, \phi \vee \langle a \rangle\, \psi) \leftrightarrow \langle a \rangle (\phi \vee \psi),$$

$$t \vdash \langle a \rangle\, \phi \wedge [a]\, \psi \rightarrow \langle a \rangle (\phi \wedge \psi),$$

$$t \vdash \langle * \rangle\, \phi \leftrightarrow \phi,$$

$$t \vdash \mathrm{ff} \langle a \rangle \leftrightarrow \mathrm{ff}, \qquad t \vdash (\phi \langle a \rangle \vee \psi \langle a \rangle) \leftrightarrow (\phi \vee \psi) \langle a \rangle,$$

$$t \vdash \phi \langle a \rangle \wedge \psi [a] \rightarrow (\phi \wedge \psi) \langle a \rangle,$$

$$t \vdash \phi \langle * \rangle \leftrightarrow \phi.$$

Rules for Recursive Assertions

$$t, \phi[\mu X.\, \phi / X] \vdash \mu X.\phi, \qquad \frac{t, \phi[\psi / X] \vdash \psi}{t, \mu X.\phi \vdash \psi}.$$

Rules for Operations

Product

The rules for the product use the projection functions π_k, for $k=0, 1$.

$$t_0 \times t_1 \vdash I \leftrightarrow I \times I,$$

$$t_0 \times t_1 \vdash \mathrm{tt} \leftrightarrow \mathrm{tt} \times \mathrm{tt}$$

$$t_0 \times t_1 \vdash \mathrm{ff} \leftrightarrow ([\mathrm{ff} \times \mathrm{tt}] \wedge [\mathrm{tt} \times \mathrm{ff}),$$

$$t_0 \times t_1 \vdash ([\phi_0 \times \phi_1] \wedge [\phi'_0 \times \phi'_1]) \leftrightarrow ([\phi_0 \wedge \phi'_0] \times [\phi_1 \wedge \phi'_1]).$$

$$t_0 \times t_1 \vdash ([\phi_0 \times \phi_1] \vee [\phi'_0 \times \phi_1]) \leftrightarrow [(\phi_0 \vee \phi'_0) \times \phi_1],$$

$$t_0 \times t_1 \vdash ([\phi_0 \times \phi_1] \vee [\phi_0 \times \phi'_1]) \leftrightarrow [\phi_0 \times (\phi_1 \vee \phi'_1)],$$

$$t_0 \times t_1 \vdash \neg [\phi_0 \times \phi_1] \leftrightarrow ([\neg \phi_0 \times \mathrm{tt}] \vee [\mathrm{tt} \times \neg \phi_1]),$$

$$t_0 \times t_1 \vdash \langle a \rangle [\phi_0 \times \phi_1] \leftrightarrow [\langle \pi_0(a) \rangle \phi_0 \times \langle \pi_1(a) \rangle \phi_1],$$

$$t_0 \times t_1 \vdash [\phi_0 \times \phi_1]\langle a \rangle \leftrightarrow [\phi_0 \langle \pi_0(a) \rangle \times \phi_1 \langle \pi_1(a) \rangle],$$

$$\frac{t_0 \vdash \phi_0,\ t_1 \vdash \phi_1}{t_0 \times t_1 \vdash \phi_0 \times \phi_1}.$$

Restriction

$$t \lceil A \vdash I \leftrightarrow I \lceil A,$$

$$t \lceil A \vdash \mathrm{tt} \leftrightarrow \mathrm{tt} \lceil A, \qquad t \lceil A \vdash \mathrm{ff} \leftrightarrow \mathrm{ff} \lceil A,$$

$$t \lceil A \vdash \neg (\phi \lceil A) \leftrightarrow (\neg \phi \lceil A),$$

$$t \lceil A \vdash (\phi_0 \lceil A) \wedge (\phi_1 \lceil A) \leftrightarrow (\phi_0 \wedge \phi_1) \lceil A,$$

$$t \lceil A \vdash (\phi_0 \lceil A) \vee (\phi_1 \lceil A) \leftrightarrow (\phi_0 \vee \phi_1) \lceil A,$$

$$t \lceil A \vdash \langle a \rangle (\phi \lceil A) \leftrightarrow (\langle a \rangle \phi) \lceil A, \qquad \text{where} \quad a \in A \cup \{*\},$$

$$t \lceil A \vdash (\phi \lceil A) \langle a \rangle \leftrightarrow (\phi \wedge R_A) \langle a \rangle \lceil A, \qquad \text{where} \quad a \in A \cup \{*\},$$

$$\frac{t \vdash R_A \to \phi}{t \lceil A \vdash \phi \lceil A}, \qquad \text{where} \quad R_A \equiv \mu X.\left(I \vee \bigvee_{\alpha \in A} X \langle \alpha \rangle\right).$$

Relabeling

$$t(\Xi) \vdash I \leftrightarrow I(\Xi),$$

$$t(\Xi) \vdash \mathrm{tt} \leftrightarrow \mathrm{tt}(\Xi), \qquad t(\Xi) \vdash \mathrm{ff} \leftrightarrow \mathrm{ff}(\Xi),$$

$$t(\Xi) \vdash \neg (\phi(\Xi)) \leftrightarrow (\neg \phi(\Xi)),$$

$$t(\Xi) \vdash \phi_0(\Xi) \wedge \phi_1(\Xi) \leftrightarrow (\phi_0 \wedge \phi_1)(\Xi),$$

$$t(\Xi) \vdash \phi_0(\Xi) \vee \phi_1(\Xi) \leftrightarrow (\phi_0 \vee \phi_1)(\Xi),$$

$$t(\Xi) \vdash \langle a \rangle(\phi(\Xi)) \leftrightarrow \Big(\bigvee_{b \in \Xi^{-1}a} \langle b \rangle \, \phi \Big)(\Xi),$$

$$t(\Xi) \vdash (\phi(\Xi))\langle a \rangle \leftrightarrow \Big(\bigvee_{b \in \Xi^{-1}a} \phi\langle b \rangle \Big)(\Xi),$$

$$\frac{t \vdash \phi}{t(\Xi) \vdash \phi(\Xi)}.$$

Sum

We use the notation j_k, $k=0, 1$, for injections on label sets. An assertion $\phi_0 + \phi_1 : t_0 + t_1$ is balanced when it satisfies $t_0 \vdash I \rightarrow \phi_0$ iff $t_1 \vdash I \rightarrow \phi_1$.

$$t_0 + t_1 \vdash I \leftrightarrow I + \mathrm{ff}, \qquad t_0 + t_1 \vdash I \leftrightarrow \mathrm{ff} + I,$$

$$t_0 + t_1 \vdash \mathrm{tt} \leftrightarrow \mathrm{tt} + \mathrm{tt}, \qquad t_0 + t_1 \vdash \mathrm{ff} \leftrightarrow \mathrm{ff} + \mathrm{ff},$$

$$t_0 + t_1 \vdash ((\phi_0 + \phi_1) \vee (\phi_0' + \phi_1')) \leftrightarrow ((\phi_0 \vee \phi_0') + (\phi_1 \vee \phi_1')),$$

$$\frac{\phi_0 + \phi_1,\ \phi_0' + \phi_1' \text{ balanced}}{t_0 + t_1 \vdash ((\phi_0 + \phi_1) \wedge (\phi_0' + \phi_1')) \leftrightarrow ((\phi_0 \wedge \phi_0') + (\phi_1 \wedge \phi_1'))},$$

$$\frac{\phi_0 + \phi_1 \text{ balanced}}{t_0 + t_1 \vdash \neg(\phi_0 + \phi_1) \leftrightarrow (\neg\phi_0) + (\neg\phi_1)},$$

$$\frac{t_1 \vdash I \rightarrow \phi_1}{t_0 + t_1 \vdash \langle j_0(a_0) \rangle(\phi_0 + \phi_1) \leftrightarrow (\langle a_0 \rangle(\phi_0 \vee I) + \mathrm{ff})} \quad \text{if} \quad a_0 \in \mathrm{sort}(t_0),$$

$$\frac{t_1 \vdash I \rightarrow \phi_1}{t_0 + t_1 \vdash (\phi_0 + \phi_1)\langle j_0(a_0) \rangle \leftrightarrow ((\phi_0 \vee I)\langle a_0 \rangle + \mathrm{ff}} \quad \text{if} \quad a_0 \in \mathrm{sort}(t_0),$$

$$\frac{t_1 \vdash I \rightarrow \neg\phi_1}{t_0 + t_1 \vdash \langle j_0(a_0) \rangle(\phi_0 + \phi_1) \leftrightarrow (\langle a_0 \rangle \, \phi_0 + \mathrm{ff})} \quad \text{if} \quad a_0 \in \mathrm{sort}(t_0),$$

$$\frac{t_1 \vdash I \rightarrow \neg\phi_1}{t_0 + t_1 \vdash (\phi_0 + \phi_1)\langle j_0(a_0) \rangle \leftrightarrow (\phi_0\langle a_0 \rangle + \mathrm{ff})} \quad \text{if} \quad a_0 \in \mathrm{sort}(t_0),$$

$$\frac{t_0 \vdash I \rightarrow \phi_0}{t_0 + t_1 \vdash \langle j_1(a_1) \rangle(\phi_0 + \phi_1) \leftrightarrow (\mathrm{ff} + \langle a_1 \rangle(\phi_1 \vee I))} \quad \text{if} \quad a_1 \in \mathrm{sort}(t_1),$$

$$\frac{t_0 \vdash I \rightarrow \phi_0}{t_0 + t_1 \vdash (\phi_0 + \phi_1)\langle j_1(a_1) \rangle \leftrightarrow \mathrm{ff} + (\phi_1 \vee I)\langle a_1 \rangle} \quad \text{if} \quad a_1 \in \mathrm{sort}(t_1),$$

$$\frac{t_0 \vdash I \rightarrow \neg \phi_0}{t_0 + t_1 \vdash \langle j_1(a_1) \rangle (\phi_0 + \phi_1) \leftrightarrow (\mathrm{ff} + \langle a_1 \rangle \phi_1)} \quad \text{if} \quad a_1 \in \mathrm{sort}(t_1),$$

$$\frac{t_0 \vdash I \rightarrow \neg \phi_0}{t_0 + t_1 \vdash (\phi_0 + \phi_1)\langle j_1(a_1) \rangle \leftrightarrow (\mathrm{ff} + \phi_1 \langle a_1 \rangle)} \quad \text{if} \quad a_1 \in \mathrm{sort}(t_1),$$

$$\frac{t_0 \vdash \phi_0, t_1 \vdash \phi_1}{t_0 + t_1 \vdash \phi_0 + \phi_1}.$$

Prefixing

$$\alpha t \vdash \mathrm{ff} \leftrightarrow \alpha \mathrm{ff},$$

$$\alpha t \vdash (\neg I \wedge \neg \alpha\phi) \leftrightarrow \alpha(\neg \phi),$$

$$\alpha t \vdash \alpha\phi_0 \wedge \alpha\phi_1 \leftrightarrow \alpha(\phi_0 \wedge \phi_1),$$

$$\alpha t \vdash \alpha\phi_0 \vee \alpha\phi_1 \leftrightarrow \alpha(\phi_0 \vee \phi_1),$$

$$\alpha t \vdash (\neg I \wedge \langle b \rangle(\alpha\phi)) \leftrightarrow \alpha(\langle b \rangle \phi), \quad \text{where} \quad b \in \mathrm{sort}(t) \cup \{*\},$$

$$\alpha t \vdash \neg \langle b \rangle I \quad \text{if} \quad * \neq b \neq \alpha,$$

$$\alpha t \vdash I \rightarrow (\langle \alpha \rangle \phi \leftrightarrow [\alpha]\phi),$$

$$\alpha t \vdash I \rightarrow \neg \langle b \rangle \phi \quad \text{if} \quad \alpha \neq b \neq *,$$

$$\alpha t \vdash (\alpha\phi)\langle b \rangle \leftrightarrow \alpha(\phi\langle b \rangle), \quad \text{where} \quad b \in \mathrm{sort}(t) \cup \{*\},$$

$$\alpha t \vdash I\langle \alpha \rangle \leftrightarrow \alpha I,$$

$$\alpha t \vdash \phi\langle b \rangle \rightarrow \neg I \quad \text{if} \quad b \neq *,$$

$$\alpha t \vdash \neg(I\langle b \rangle) \quad \text{if} \quad * \neq b \neq \alpha,$$

$$\alpha t \vdash I \rightarrow \langle a \rangle(\alpha I),$$

$$\frac{t \vdash \phi \rightarrow \phi'}{\alpha t \vdash \alpha\phi \rightarrow \alpha\phi'}.$$

Looping

$$(t/\alpha, J) \vdash \mathrm{tt} \leftrightarrow \mathrm{tt}/\alpha, J,$$

$$(t/\alpha, J) \vdash \mathrm{ff} \leftrightarrow \mathrm{ff}/\alpha, J,$$

$$(t/\alpha, J) \vdash I \leftrightarrow I/\alpha, J,$$

$$(t/\alpha, J) \vdash \phi_0/\alpha, J \wedge \phi_1/\alpha, J \leftrightarrow (\phi_0 \wedge \phi_1)/\alpha, J,$$

$$(t/\alpha, J) \vdash \phi_0/\alpha, J \vee \phi_1/\alpha, J \leftrightarrow (\phi_0 \vee \phi_1)/\alpha, J,$$

$$(t/\alpha, J) \vdash (\neg \phi)/\alpha, J \leftrightarrow \neg(\phi/\alpha, J),$$

$$(t/\alpha, J) \vdash \langle b \rangle(\phi/\alpha, J) \leftrightarrow (\langle b \rangle\, \phi)/\alpha, J \qquad \text{if} \quad b \neq \alpha,$$

$$\frac{t \vdash I \rightarrow \neg\phi}{(t/\alpha, J) \vdash \langle \alpha \rangle(\phi/\alpha, J) \leftrightarrow (\langle \alpha \rangle\, \phi)/\alpha, J},$$

$$\frac{t \vdash I \rightarrow \phi}{(t\alpha, J) \vdash \langle \alpha \rangle(\phi/\alpha, J) \leftrightarrow (\langle \alpha \rangle\, \phi \vee J)/\alpha, J},$$

$$(t/\alpha, J) \vdash (\phi/\alpha, J)\langle b \rangle \leftrightarrow (\phi\langle b \rangle)/\alpha, J \qquad \text{if} \quad b \neq \alpha,$$

$$\frac{t \vdash J \rightarrow \neg\phi}{(t/\alpha, J) \vdash (\phi/\alpha, J)\langle \alpha \rangle \leftrightarrow (\phi\langle \alpha \rangle)/\alpha, J},$$

$$\frac{t \vdash J \rightarrow \phi}{(t/\alpha, J) \vdash (\phi/\alpha, J)\langle \alpha \rangle \leftrightarrow (\phi\langle \alpha \rangle \vee I)/\alpha, J},$$

$$\frac{t \vdash \phi}{(t/\alpha, J) \vdash \phi/\alpha, J}.$$

nil : $\quad \text{nil} \vdash I$

Acknowledgments

The use of indexed category theory to incorporate labelling into the categorical presentation of models for parallel computation was originally suggested by Mike Fourman, who proposed that restriction and hiding could be viewed as some form of generalised universal and existential quantification. This line was followed up by Valeria de Paiva, Edmund Robinson, and Pino Rosolini, who uncovered some diffculties in fulfilling this suggestion for Petri nets but checked that with partial labelling functions one obtained a fibration with functors λ^* with left adjoint $\lambda_!$ between fibres. (Thanks to Edmund for writing it down.) Their intuition was that an event with an undefined label was an internal or hidden event. However, this does not square with the use of products to derive parallel compositions. And because I could see no other way of deriving parallel compositions as categorical constructions, I have used what amounts to total labelling functions on events. I have benefitted from helpful remarks and encouragement from Thierry Coquand, Martin Hyland, and Mogens Nielsen. Paul Taylor's thesis (1987) and Peter Johnstone's course notes (1983) provided useful sources of indexed category theory. I am grateful to Anders Kock for remarks which led to several improvements. Finally, Yuri Gurevich and the anonymous referee are to be thanked for not allowing me to get away with things too easily.

RECEIVED April 1, 1989; FINAL MANUSCRIPT RECEIVED August 15, 1989

References

ABRAMSKY, S. (1987), Domain theory in logical form, LIC 1987.

BÉNABOU, J. (1985), Fibred categories and the foundations of naive category theory, *J. Symbolic Logic* **50**.

BROOKES, S. D. (1983), On the relationship of CCS and CSP, *in* "Proceedings, ICALP 1983," Lecture Notes in Computer Science, Vol. 154, Springer-Verlag, New York/Berlin.

BROOKES, S. D. (1985), On the axiomatic treatment of concurrency, *in* "Lecture Notes in Computer Science, Vol. 197," Springer-Verlag, New York/Berlin.

CAMPBELL, R. H., AND HABERMANN, A. N. (1973), The specification of process synchronisation by path expressions, *in* Lecture Notes in Computer Science, Vol. 16," Springer-Verlag, New York/Berlin.

GRAF, I., AND SIFAKIS, J. (1985), "A Logic for the Description of Nondeterministic Programs and Their Properties," Report IMAG RR 551, Grenoble, France.

HOARE; C. A. R., BROOKES, S. D., AND ROSCOE, A. W. (1984), A theory of communicating processes, *J. Assoc. Comput. Mach.*

HOARE, C. A. R., AND OLDEROG, E. (1983), Specification oriented semantics for communicating processes, *in* "Lecture Notes in Computer Science, Vol. 154," Springer-Verlag, New York/Berlin.

KOZEN, D. (1982), Results on the propositional μ-calculus, *in* "Proceedings, ICALP 1982".

LABELLA, A., AND PETTEROSSI, A. (1985), Categorical models of process cooperation, *in* "Lecture Notes in Computer Science, Vol. 240," Springer-Verlag, New York/Berlin.

LAUER, P., AND CAMPBELL, R. H. (1974), Formal semantics for a class of high level primitives for coordinating concurent processes, *Acta Inform.* **5**.

MACLANE, S. (1971), "Categories for the Working Mathematician," Graduate Texts in in Mathematics, Springer-Verlag, New York/Berlin.

MARTIN-LÖF, P. (1983), The domain interpretation of type theory, Lecture notes, Göteborg.

MILNER, A. R. G. (1980), Calculus of communicating systems, *in* "Lecture Notes in Computer Science, Vol. 92," Springer-Verlag, New York/Berlin.

MILNER, A. R. G. (1989), "Communication and Concurrency," Prentice–Hall, Englewood Cliffs, NJ.

POIGNÉ, A. (1985), Category theory and logic, *in* "Lecture Notes in Computer Science, Vol. 240," Springer-Verlag, New York/Berlin.

PRATT, V. R. (1986), Modelling concurrency with partial orders, *Internat. J. Parallel Programming* **15**, No. 1 (February) 33–71.

STIRLING, C. (1985), A complete modal proof system for a subset of SCCS, *in* "Lecture Notes in Computer Science, Vol. 185," Springer-Verlag, New York/Berlin.

TAYLOR, P. (1987), "Recursive Domains, Indexed Category Theory and Polymorphism," Ph. D. thesis, Cambridge University.

WINSKEL, G. (1982), Event structure semantics of CCS and related languages, *in* "Proceedings, ICALP 1982," Lecture Notes in computer Science, Vol. 140, Springer-Verlag, New York/Berlin.

WINSKEL, G. (1984), Categories of Models for concurrency, *in* "Lecture Notes in Computer Science, Vol. 197," Springer-Verlag, New York/Berlin.

WINSKEL, G. (1985a), A complete proof system for SCCS with modal assertions, *in* "Lecture Notes in Computer Science, Vol. 206," Springer-Verlag, New York/Berlin.

WINSKEL, G. (1985b), Synchronisation trees, *TCS* (May).

WINSKEL, G. (1987a), Petri nets, algebras, morphisms and compositionality, *Inform. and Comput.* (March).

WINSKEL, G. (1987b), Event structures, *in* "Lecture Notes in Computer Science, Vol. 255," Springer-Verlag, New York/Berlin.

WINSKEL, G. (1988), A category of labelled Petri nets and compositional proof systems, *in* "Proceedings, Conference on Logic in Computer Science, Edinburgh, July."